Departament de Teoria del Senyal i Comunicacions

Universitat Politècnica de Catalunya

Ph.D. Thesis

Contribution to Dimensionality Reduction of Digital Predistorter Behavioral Models for RF Power Amplifier Linearization

Author
Thi Quynh Anh Pham

Advisors
Dr. Pere Lluis Gilabert Pinal
Dr. Gabriel Montoro López

Barcelona
November 6, 2019

Abstract

The power efficiency and linearity of radio frequency (RF) power amplifiers (PAs) are critical in wireless communication systems. The main scope of PA designers is to build the RF PAs capable to maintain high efficiency and linearity figures simultaneously. However, these figures are inherently conflicted to each other and system-level solutions based on linearization techniques are required.

Digital predistortion (DPD) linearization has become the most widely used solution to mitigate the efficiency versus linearity trade-off. The dimensionality of the DPD model depends on the complexity of the system. It increases significantly in high efficient amplification architectures when considering current wideband and spectrally efficient technologies. Overparametrization may lead to an ill-conditioned least squares (LS) estimation of the DPD coefficients, which is usually solved by employing regularization techniques. However, in order to both reduce the computational complexity and avoid ill-conditioning problems derived from overparametrization, several efforts have been dedicated to investigate dimensionality reduction techniques to reduce the order of the DPD model.

This dissertation contributes to the dimensionality reduction of DPD linearizers for RF PAs with emphasis on the identification and adaptation subsystem. In particular, several dynamic model order reduction approaches based on feature extraction techniques are proposed. Thus, the minimum number of relevant DPD coefficients are dynamically selected and estimated in the DPD adaptation subsystem. The number of DPD coefficients is reduced, ensuring a well-conditioned LS estimation while demanding minimum hardware resources. The presented dynamic linearization approaches are evaluated and compared through experimental validation with an envelope tracking PA and a class-J PA. The experimental results show similar linearization performance than the conventional LS solution but at lower computational cost.

Contents

Acronyms

ACEPR	Adjacent Channel Error Power Ratio
ACLR	Adjacent Channel Leakage Ratio
ACPR	Adjacent Channel Power Ratio
ADC	Analog to Digital Converter
AIC	Akaike Information Criterion
AM	Amplitude Modulation
APCA	Adaptive Principal Component Analysis
APD	Analog Predistortion
ARM	Advanced RISC Machines
BB	Base Band
BD-APCA	Block-deflated APCA
BIC	Bayesian Information Criterion
BS	Base Station
BW	Bandwidth
Capex	Capital expenditure
CCA	Canonical Correlation Analysis
CDMA	Code-Division Multiple Access
CFR	Crest Factor Reduction
CFS	Correlation-based Feature Selection
CGHA	Complex domain Generalized Hebbian Algorithm
CH	Channel
CS	Compressed Sensing
DAC	Digital to Analog Converter
DB	Dual Band
DDR	Dynamic Deviation Reduction
DOTM	Dynamic Orthonormal Transformation Matrix algorithm
DPD	Digital Predistortion
DPLS	Dynamic Partial Least Squares
DUT	Device Under Test
EA	Envelope Amplifier

EDGE	Enhanced Data rates for GSM Evolution
EER	Envelope Elimination and Restoration
EMP	Envelope Memory Polynomial
ET	Envelope Tracking
E-UTRA	Evolved Universal Terrestrial Radio Access
EVM	Error Vector Magnitude
FBMC	Filter Bank Multi-Carrier
FCBF	Fast Correlated based Filter
FC-FBMC	Fast-Convolution Filter Bank Multi-Carrier
FM	Frequency Modulation
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
GFDM	Generalized Frequency Division Multiplexing
GMP	Generalized Memory Polynomial
GPS	Global Positioning System
HBT	Heterojunction Bipolar Transistor
HMP	Hybrid Memory Polynomial
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency
IoT	Internet of Things
IQ MOD	In-phase and Quadrature-phase Modulator
LARS	Least Angle Regression
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LHC	Lawson-Hanson-Chan (bidiagonalization)
LINC	Linear Amplification using Nonlinear Components
LMBA	Load Modulated Balanced Power Amplifier
LO	Local Oscillator
LPA	Linear Power Amplifier
LRBR	Less Relevant Basis Removal
LS	Least Squares
LTE	Long-Term Evolution
LUT	Look-up Table

M-QAM	M-ary Quadrature Amplitude Modulation
MAF	Masked Autoregressive Flow
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MMS	Multi Media Service
MP	Memory Polynomial
NLPA	Nonlinear Power Amplifier
NMSE	Normalized Mean Square Error
OFDM	Orthogonal Frequency-Division Multiplexing
OLS	Ordinary Least Squares
OMP	Orthogonal Matching Pursuit
OMP-BIC	The combination of OMP and BIC
OMP-col	The original OMP algorithm
OMP-LUT	The OMP algorithm applies for the basis presented in LUTs
Opex	Operating cost/expenditure
PA	Power Amplifier
PAE	Power Added Efficiency
PAPR	Peak-to-Average Power Ratio
PCA	Principal Components Analysis
PDSCH	Physical Downlink Shared Channel
PLS	Partial Least Squares
PM	Phase Modulation
PS	Processing System
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
QR	QR factorization
R&S	Rohde & Schwarz
RC	Radio Control (system)
RF	Radio Frequency
RLS	Recursive Least Squares
RPCA	Robust Principal Component Analysis
Rx	Receiver
SBL	Sparse Bayesian Learning

SFS	Sequential Forward Selection
SIMO	Single Input Multiple Output
SISO	Single Input Single Output
SMS	Short Message Service
SoC	System on Chip
SPSA	Simultaneous Perturbation Stochastic Approximation (behavioral model)
SSA	Signal and Spectrum Analyzer
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TDMA	Time Division Multiple Access
TI	Texas Instruments
TWA	Traveling Wave Power Amplifier
Tx	Transmitter
UFMC	Universal Filtered Multi-Carrier
Vdd	Drain-to-drain voltage
Vds	Drain-to-source voltage
Vgs	Gate-to-source voltage
VSG	Vector Signal Generator
WCDMA	Wide-band CDMA
WCS	Wireless Communication System

Chapter 1

Introduction

1.1 Motivation

Radio frequency (RF) power amplifiers (PAs) are crucial components in wireless communications systems (WCSs). They are the most power consuming devices in the transmitter chain and one of the main sources of nonlinear distortion. Figure 1.1 shows a survey that demonstrates the significance in power consumption [1]: the base station accounts for 57% of the total supplied energy in a cellular system, in which, 50-80% of this amount is distributed to the PAs. Therefore, the PA power efficiency is a key feature to lower the running costs. The PA achieves highest power efficiency when operated close to saturation, but also tends to be nonlinear. Figure 1.2 shows this the compression in the input-output characteristics for higher output levels. Nonlinearity causes effects such as distortion in the output signal and channel interference.

The increasing demand for high-capacity services and applications requires high speed data rates. This sets the path towards the every-time-more-spectrally-efficient modulation schemes (high M-QAM schemes) and access techniques, such as wide-band code-division multiple access (WCDMA) in 3G, orthogonal-frequency-division multiplexing (OFDM) in 4G long term evolution (4G-LTE) and 5G New Radio (5G-NR), or filter bank multi-carrier (FBMC) - a discarded candidate waveform for 5G-

NR. The complex nonconstant envelope modulated signals generated by such modulation schemes and access techniques present high peak-to-average-power ratios (PAPRs), which forces the PA to operate with significant back-off levels in order to avoid

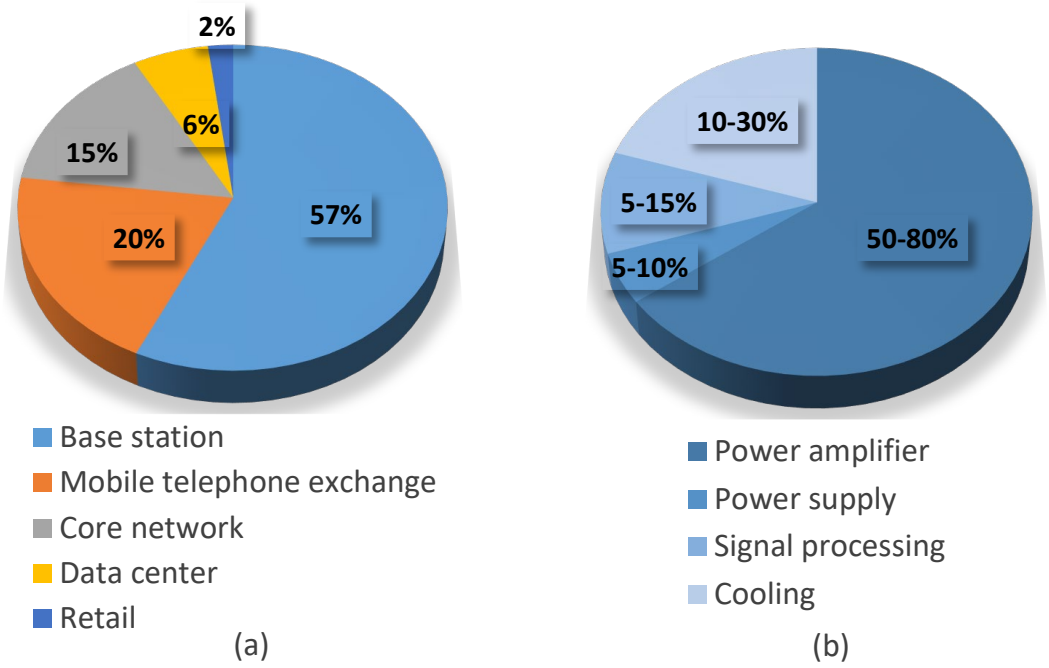


Figure 1.1: Power consumption distribution in (a) a wireless cellular system and (b) a cellular base station. [1].

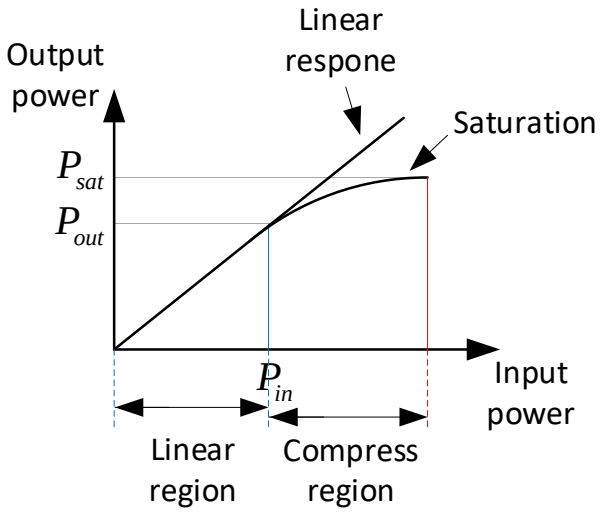


Figure 1.2: Nonlinear characteristic of a RF PA.

nonlinear distortion with the consequent degradation in average power efficiency. In order to avoid wasting energy resources when handling these high PAPR signals, highly efficient amplification architectures, such as outphasing [2], Doherty [3], envelope tracking (ET) [4], envelope elimination and restoration (EER) [5] and load modulated balanced amplifiers (LMBA) [6] have been adopted. All these high efficient PA architectures demand the use of linearization techniques to guarantee the required linearity levels, such as, for example, feedforward [7], feedback [8] and predistortion (digital [9] and analog [10]) linearization. Among them, digital predistortion (DPD) linearization is the most extended solution to cope with the linearity versus power efficiency trade-off due to its flexibility and good linearization performance.

The linearization of high efficient PAs that amplify spectrally efficient wideband signals with high PAPR ($>$ 10 dB) is challengeable due to the fact that the DPD behavioral model may require a significantly high number of coefficients to meet the stringent linearity specifications [11]. Increasing the number of coefficients, not only derives in an increase of the computational complexity, but also may lead to uncertainty of the least squares (LS) coefficient estimation. Therefore, it is essential to keep the number of coefficients of the DPD model to the minimum necessary.

This thesis presents and discusses three dimensionality reduction techniques that are aimed to reduce the computational complexity of DPD linearizers by:

- **dynamically reducing the number of DPD coefficients to be estimated** and
- **simplifying the DPD coefficient estimation and adaptation process**.

To the best author's knowledge, for the first time, dynamic linearization approaches oriented at the simplification of the DPD coefficient estimation by taking advantage of the orthogonality property among the transformed basis functions of the DPD behavioral model are presented. These techniques are validated by means of two different experimental test beds: one located at the laboratory of the Components

and Systems for Communications research group and the other located at the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC).

1.2 Outline of Thesis

The dissertation presents different dimensionality reduction techniques for DPD linearization of RF PAs and their effects on the DPD system which are organized as follows.

First, Chapter 1 presents the motivation of the work, the outline of the dissertation and a list of research publications related to the work developed by the Ph.D candidate.

Chapter 2 analyzes the linearity versus efficiency trade-off of PAs, then presents an overview on high-efficiency amplification topologies and linearization techniques to cope with the trade-off.

Chapter 3 includes a brief introduction to PA behavioral modeling, the principles of DPD linearization, numerical methods for solving the least squares solution and specific details on look-up table (LUT) implementation of DPD behavioral models.

Chapter 4 presents an overview of dimensionality reduction techniques for PA behavioral modeling and DPD linearization, focusing on feature selection techniques for the DPD forward path and feature extraction techniques for the DPD observation path.

The next three Chapters describe sequently the new approaches for PA behavioral modeling and DPD coefficient estimation/adaptation. Chapter 5 focuses on the dimensionality reduction technique based on the principal component analysis (PCA), named block-deflated adaptive PCA (BD-APCA) which allows to reduce the number of estimated DPD coefficients, estimate the coefficients independently and enhance the robustness of the estimation.

Chapter 6 presents the estimation/adaptation of DPD coefficients employing two

dimensionality reduction techniques, namely partial least squares (PLS) and principal component analysis, and providing a comparison between their linearization performances. Subsequently, a new technique based on PLS, named dynamic partial least squares (DPLS), is introduced to dynamically select, at each iteration of the DPD adaptation, the minimum number of components required to guarantee a certain identification performance of the DPD coefficients.

Chapter 7 describes a dimensionality reduction method, named PCA-DPLS, that is a combination of PCA (computed off-line) and DPLS. The combination allows dynamic DPD estimation/adaptation achieving similar dimensionality reduction capabilities than with Canonical Correlation Analysis (CCA is the method that shows better model order reduction capabilities in comparison to PCA and PLS) but with lower computational cost.

Finally, Chapter 8 gives the conclusion on the dissertation and discusses possible future research lines in the field of DPD linearization. As a summary, Figure 1.3 depicts the main content of this dissertation.

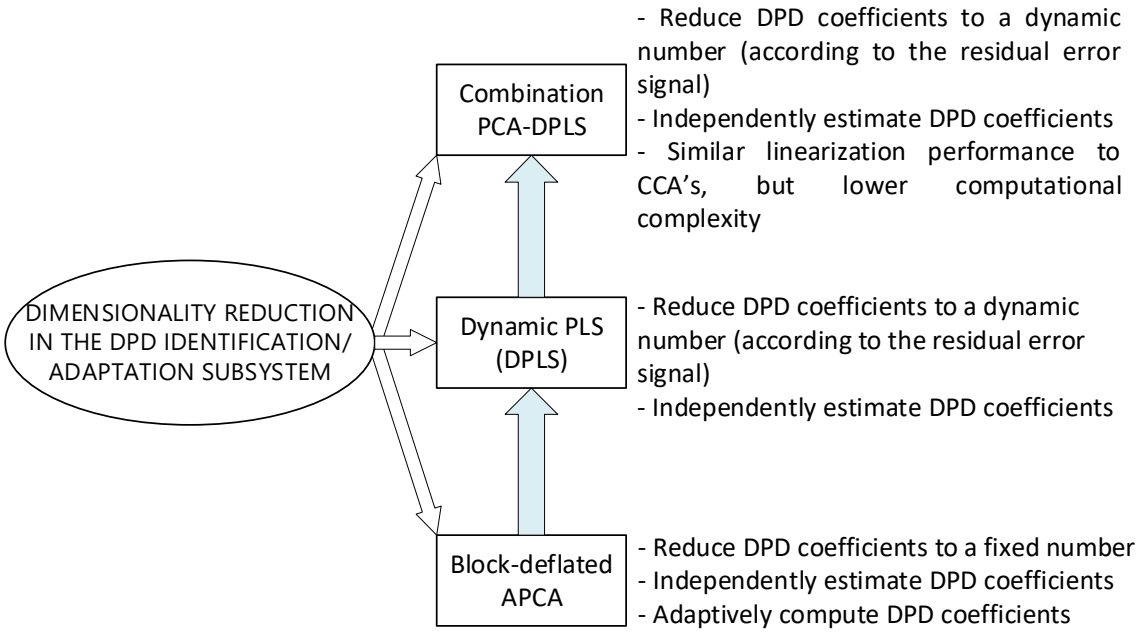


Figure 1.3: Overview of main Chapters of this dissertation.

1.3 Research Contributions

This Ph.D. dissertation contributes to the high power amplifier linearization using DPD technique. Dynamic linearization approaches are proposed to dynamically determine the number of required DPD coefficients to be estimated/updated in DPD adaptation subsystem. The approaches reduce the computational complexity of the DPD linearization system, at the same time, avoid the overfitting and uncertainty of LS estimation. The experimental tests have been carried out to validate the performance of these techniques.

The research reported in this thesis has generated publications in international conferences and journal papers. The publications are listed in the following.

For block-deflated adaptive principal component analysis (Chapter 5):

- Q. A. Pham, D. López-Bueno, T. Wang, G. Montoro, and P. L. Gilabert, "Multidimensional LUT-based digital predistorter for concurrent dual-band envelope tracking power amplifier linearization," in Proc. 2018 IEEE Topical Conf. on 93 RF/Microw. Power Amplifiers for Radio and Wireless Appl. (PAWR), Jan. 2018, pp. 47-50.

- Q. A. Pham, D. López-Bueno, G. Montoro, and P. L. Gilabert, "Adaptive principal component analysis for online reduced order coefficient extraction in PA behavioral modeling and DPD linearization," in 2018 IEEE MTT-S Int. Microw. Symp. (IMS), Jun. 2018, pp. 160-163.

- D. López-Bueno, Q. A. Pham, G. Montoro, and P. L. Gilabert, "Independent digital predistortion coefficients estimation using adaptive principal component analysis," IEEE Transactions on Microwave Theory and Techniques, vol. 66, no. 12, pp. 5771-5779, Dec. 2018.

For partial least squares and dynamic partial least squares (Chapter 6):

- Q. A. Pham, D. López-Bueno, T. Wang, G. Montoro, and P. L. Gilabert, "Partial least squares identification of multi look-up table digital predistorters for concurrent dual-band envelope tracking power amplifiers," IEEE Transactions on Microwave Theory and Techniques, vol. 66, no. 12, pp. 5143-5150, Dec. 2018.

- Q. A. Pham, D. López-Bueno, G. Montoro, and P. L. Gilabert, "Dynamic selection and update of digital predistorter coefficients for power amplifier linearization," in Proc. 2019 IEEE Topical Conf. on RF/Microw. Power Amplifiers for Radio and Wireless Appl. (PAWR), Jan. 2019.

For combination of principal component analysis and dynamic partial least squares (Chapter 7):

- Q. A. Pham, G. Montoro, D. López-Bueno, and P. L. Gilabert, "Dynamic selection and estimation of the digital predistorter coefficients for power amplifi er linearization," IEEE Transactions on Microwave Theory and Techniques, vol. 67, no. 10, pp. 3996-4004, Oct. 2019.

Book Chapter:

- P. L. Gilabert, D. López-Bueno, and Q. A. Pham, G. Montoro, "Chapter 17: Machine Learning for Digital Front-End: a Comprehensive Overview," in Book Fa-Long Luo., "Machine Learning for Future Wireless Communications", John Wiley & Sons, Inc., Hoboken, New Jersey, (**accepted and to be published in 2020**).

Chapter 2

Linearity and Power Efficiency Issues in Power Amplification

The RF PA is an important element in the transmitters in WCS base stations. Linearity and power efficiency are the key factors in power amplification. Linearity ensures the accuracy of the amplification, whereas, high power efficiency enables to reduce the power consumption and the size of the cooling system at the base station thus minimizing the operating cost. In the mobile handset, high efficiency PA lengthens the battery lifetime. Unfortunately, linearity and power efficiency are inherently conflicted to each other. For that reason, a lot of effort has been dedicated to find efficient amplification architectures which are properly combined with system level linearizers and capable to mitigate this trade-off. This Chapter presents the PA linearity-efficiency compromise in Section 2.1, the methods to improve power efficiency in Section 2.2 and the methods to preserve the required linearity levels in Section 2.3.

2.1 Linearity versus Efficiency Trade-off

Ideally, a linear power amplifier produces its output voltage as a scalar multiple of its input voltage. Considering a memoryless linear PA (so that the memory effects can

be ignored), its amplification can be expressed as

$$v_{out}(t) = T[v_{in}] = g v_{in}(t) \tag{2.1}$$

with $v_{in}(t)$ be the amplifier's input voltage, $v_{out}(t)$ the amplifier's output voltage, $T[\cdot]$ the transfer function and g the scalar voltage gain. In the ideal linear amplification, the PA's output and input are identical (except for the scalar gain), no additional in-band or out-of-band frequency components are introduced.

However, in practice, the PA presents a nonlinear behavior when approximating towards compression, as shown in Figure 1.2. The nonlinearity can be caused not only by the amplifier itself but also by the oscillators or the mixers. In amplitude and phase modulated signals, the nonlinear distortion introduced by PAs appears as spectral regrowth. Figure 2.1 shows the spectral regrowth that appears in the spectrum of the amplified output signal (red) with respect to the spectrum of the input signal (blue) due to the PA nonlinear behaviour.

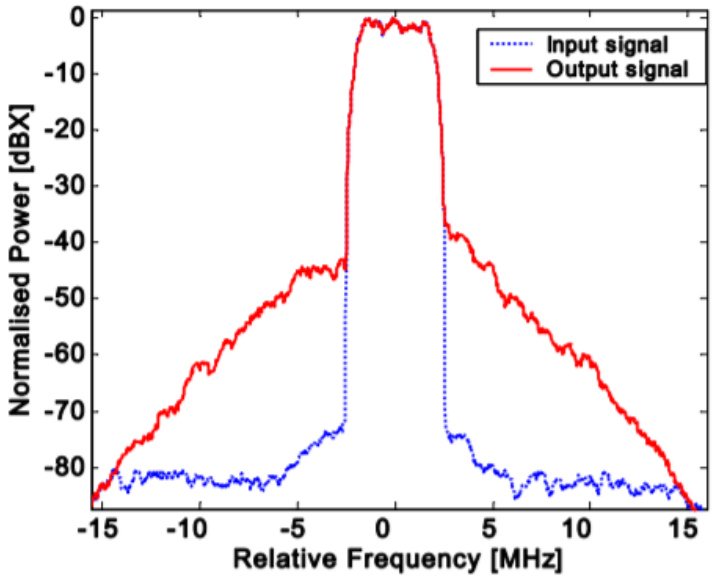


Figure 2.1: Power spectral density of the input and output signals of a nonlinear power amplifier using a WCDMA signal [12].

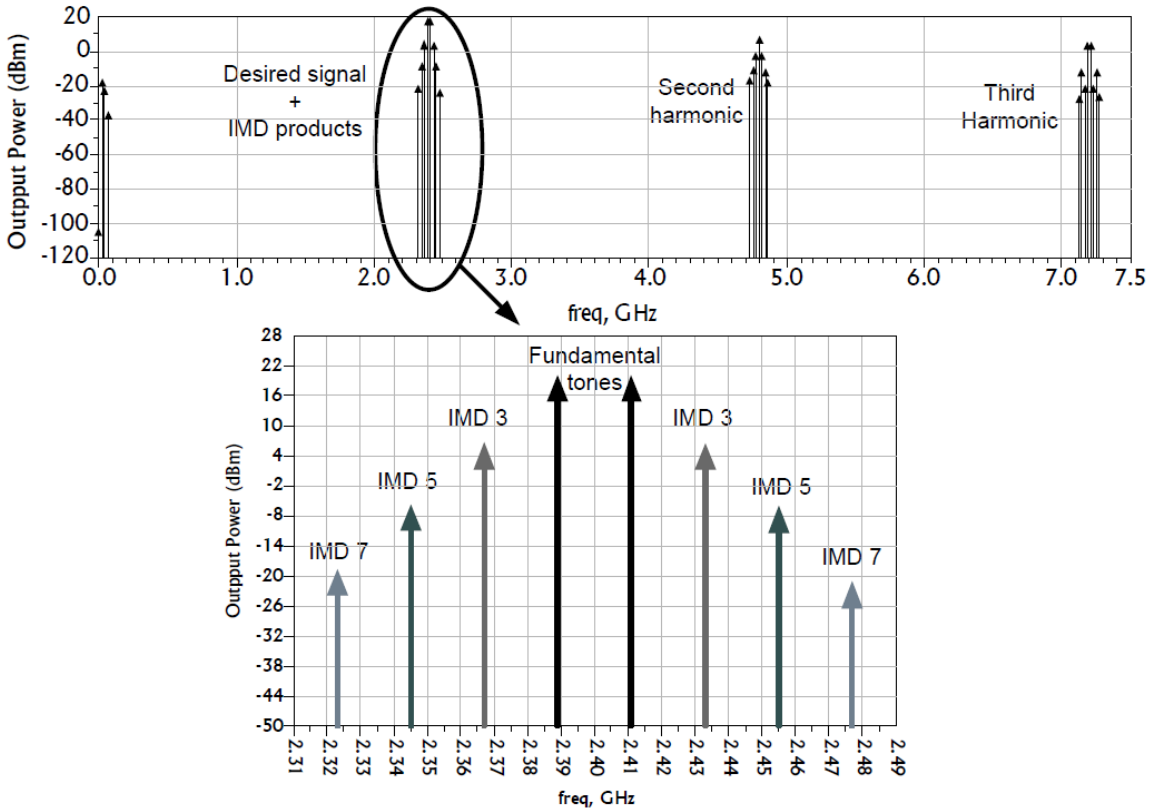


Figure 2.2: Harmonic and intermodulation distortion at the output of a nonlinear PA considering a two-tone test [13].

The output signal $v_{out}(t)$ of a nonlinear PA is a nonlinear function of its input signal $v_{in}(t)$ which can be expressed as

$$v_{out}(t) \approx \sum_{k=1}^{\infty} g_k v_{in}^k(t). \tag{2.2}$$

The polynomial function in (2.2) includes a series of terms proportional to $v_{in}(t)$, in which g_k is the voltage gain of each term in the series. The first term of this series corresponds to the linear term and is the desired output signal. The rest of the terms present additional frequency components. Specifically, even order terms of the polynomial series $(v_{in}^2, v_{in}^4, .., v_{in}^{2k})$ describe the integer multiples of the input signal (harmonics). The nonlinear distortion introduced by even order terms is named

harmonic distortion (HD). Whereas, odd order terms ($v_{in}^3, v_{in}^5, .., v_{in}^{2k-1}$) show the intermodulation distortion (IMD) that occurs when the input signal contains more than one different frequencies. In case some odd order terms fall inside the signal bandwidth, they cause the in-band distortion. Figure 2.2 shows the harmonic and intermodulation distortion in the output of the nonlinear PA when considering a two-tone test.

The distortion measurement of the PA output signal can be carried out using the amplitude modulation to amplitude modulation (AM-AM) and amplitude modulation to phase modulation (AM-PM) characteristics. The linearity of the PA can be characterized with several figures of merit, such as the error vector magnitude (EVM) and adjacent channel power ratio ($ACPR$).

To quantify the in-band distortion, the EVM measures the effects of the distortion on the amplitude and phase (I and Q) of the modulated output signal. The error vector (see Figure 2.3) describes dissimilarity between the measured IQ modulation signal and a reference (ideal) signal. The EVM is defined as the square root of the ratio between the mean error vector power $S_{err} = \frac{1}{N}\sum_{1}^{N}(\Delta I^2 + \Delta Q^2)$ to the mean reference (ideal) vector power $S_{ref} = \frac{1}{N}\sum_{1}^{N}\left(I_{ref}^2 + Q_{ref}^2\right)$, N is the number of the

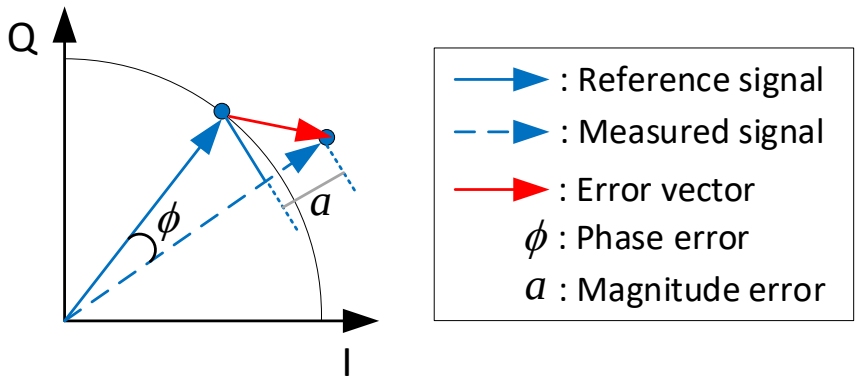


Figure 2.3: Error vector magnitude representation.

samples (of the constellation) of the signals.

$$EVM = \sqrt{\frac{S_{err}}{S_{ref}}}\ [\%] \tag{2.3}$$

The figure of merit $ACPR$, also known as the adjacent channel leakage power ratio ($ACLR$), measures the spectral regrowth present in the spectrum of the PA output signal. $ACPR$ is defined as the ratio of the power delivered in the adjacent channel (upper or lower sideband) to the total power over the channel bandwidth [14].

$$ACPR = 10\log_{10}\left(\frac{P_{adjacent-channel}}{P_{main-channel}}\right) = 10\log_{10}\left(\frac{\int\limits_{adj} P_{out}(f)\cdot df}{\int\limits_{chan} P_{out}(f)\cdot df}\right)\ [dB] \tag{2.4}$$

Communication standards specify the maximum allowed out-of-band power emission in terms of a spectrum emission mask and $ACPR$. Similarly to the spectral regrowth limitations, communications standards determine maximum levels of the EVM permitted at the transmitter antenna and at the receiver, depending on the modulation scheme used and the codification (optional).

The efficiency of a PA (η) shows the capability of the PA to convert dc power from the supply into RF energy of the output signal. It is calculated as the percentage of the output power of PA (P_{out}) to the dc power taken from the supply (P_{dc}) (see Figure 2.4):

$$\eta = \frac{P_{out}}{P_{dc}}[\%]. \tag{2.5}$$

Besides, the power added efficiency (PAE) that measures the efficiency of the PA considering the already existing power from input signal is defined as

$$PAE = \frac{P_{out} - P_{in}}{P_{dc}}[\%]. \tag{2.6}$$

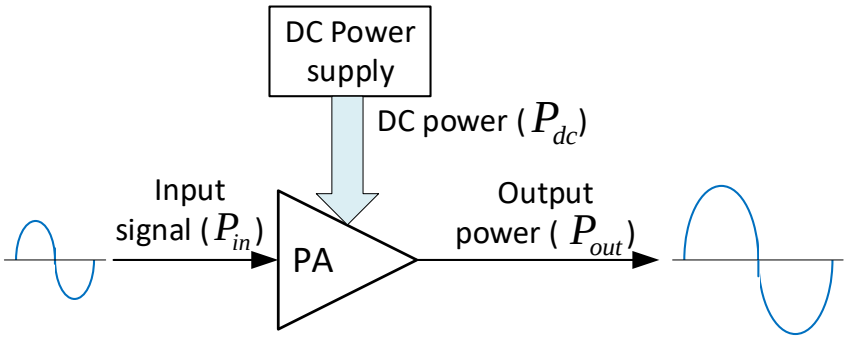


Figure 2.4: Block diagram of the PA with power supply.

In a system that uses the nonconstant envelope signals, the PA *instantaneous efficiency*, defined as the efficiency at a specific output power, is higher at the peaks and lower at the non-peaks of the output signal. In other words, signals with time-varying amplitudes produce time-varying efficiencies. The average efficiency (η_{AVG}) is to measure the performance of the time-varying signal amplification. η_{AVG} is defined as the ratio of the average output power to the average dc power.

$$\eta_{AVG} = \frac{P_{outAVG}}{P_{dcAVG}}[\%]. \tag{2.7}$$

When dealing with modern spectrally efficient multi-carrier (e.g., OFDM-based) waveforms presenting high PAPR, the PA needs to operate at large power back-off leading to a serious degradation of average efficiency. As shown in Figure 2.5, the PA has to operate with significant back-off from the 1 dB compression point to prevent the peaks of the signal going into compression. Consequently, the average PAE of the amplification is low. Note that the PA efficiency not only depends on the back-off level chosen to operate the PA but also on the PA operation class [15].

The combination of high efficient amplification topologies with linearization techniques can overcome or at least mitigate the linearity-efficiency trade-off [7]. High efficient amplification architectures based on dynamic load (e.g. outphasing and Doherty) or dynamic supply modulation (e.g. ET and EER) are adopted to enhance

the power efficiency figures when dealing with nonconstant envelope signals that have high PAPRs. Additionally, system level linearization techniques such as, feedforward, feedback and predistortion are employed in order to guarantee the required linearity levels.

2.2 High-efficiency Amplification Topologies

This Section presents the popular topologies for high efficiency amplification: outphasing, Doherty, ET and EER PAs. Outphasing and Doherty PAs make use of load modulation techniques combining the operation of two or more PAs. While ET and EER topologies improve the power efficiency through dynamic supply modulation

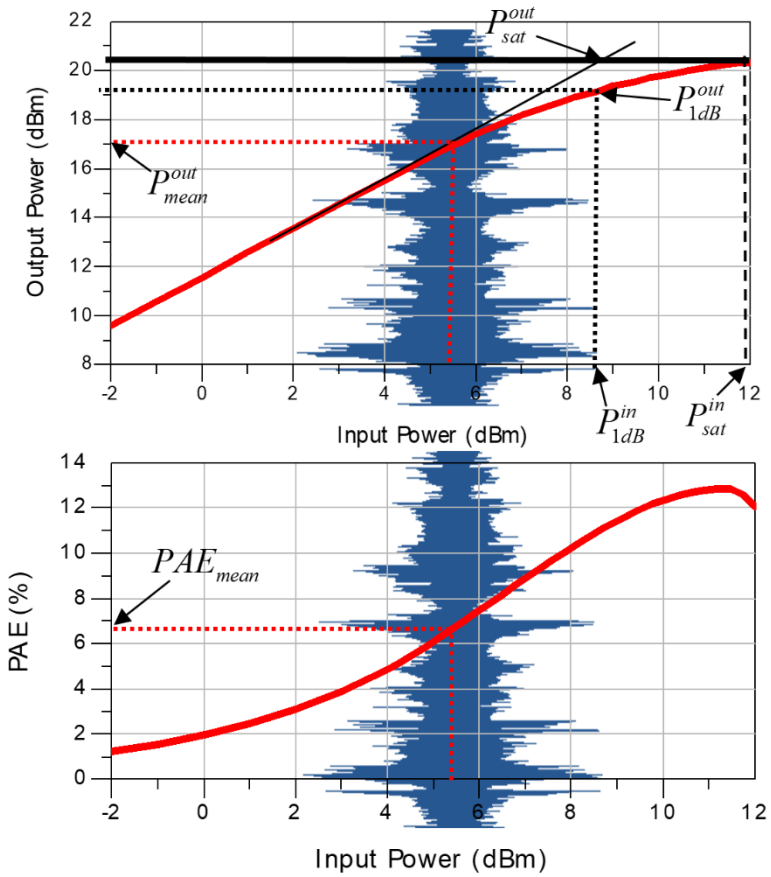


Figure 2.5: Linearity versus power efficiency trade-off.

techniques that do not require additional RF PAs, but require high efficient envelope modulators.

2.2.1 Dynamic Load-Modulation Techniques

Outphasing

The outphasing modulation architecture was invented by H. Chireix in 1935 [2], and later revised and re-introduced under the name of linear amplification with nonlinear components (LINC) by D. Cox in 1974 [16]. The outphasing architecture consists in amplifying the amplitude-modulated signal by combining the amplification of two equal constant amplitude phase-modulated signals. This scheme can be conducted as following. First, the signal component separator splits the input amplitude-modulated signal $S(t) = A(t)\cos\left[\omega_c t + \varphi(t)\right]$ ($A(t)$ is the amplitude, ω_c the carrier frequency and $\varphi(t)$ the phase shift of $S(t)$) into two constant-envelope phase-modulated signals with opposite phases as

$$\begin{aligned} S(t) =& S_1(t) + S_2(t) \\ =& \frac{A_{max}}{2}\cos(\omega_c t + \varphi(t) + \phi(t)) + \\ & \frac{A_{max}}{2}\cos(\omega_c t + \varphi(t) - \phi(t)) \end{aligned} \tag{2.8}$$

where A_{max} be the maximum amplitude of $S(t)$ and $\phi = cos^{-1}\left(\frac{A(t)}{A_{max}}\right)$ be half the outphasing angle. Then the two branch signals $S_1(t)$ and $S_2(t)$ are independently amplified by two separate amplifiers. Since $S_1(t)$ and $S_2(t)$ have constant envelopes, the two branch PAs can operate in high efficient switched mode. At the PA outputs, the amplified signals are recombined to get the amplified replica of the original signal. The high-level architecture and operation principle of the outphasing PA is shown in Figure 2.6.

There are two popular techniques of outphasing: LINC and Chireix outphasing.

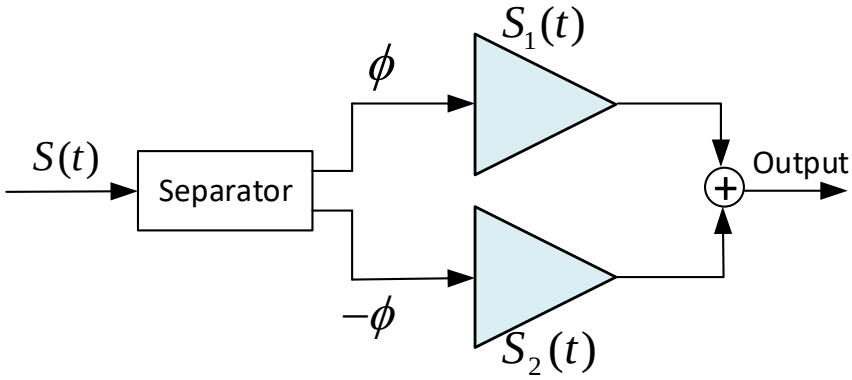


Figure 2.6: Simplified outphasing configuration.

The main point that makes these techniques distinct from each other is the type of employed power combiner: isolated or non-isolated. LINC uses an isolated combiner while Chireix outphasing has a non-isolated one. The difference in the combiner affects to the system operation, resulting different linearity and efficiency enhancement. The Chireix outphasing with the non-isolate combiner gains good efficiency but not good linearity. Whereas, the LINC with isolate combiner achieves good linearity but lower power efficiency [17].

Although the outphasing gains high efficiency, some drawbacks of this topology can be pointed out as follows:

- The signal separation require complex calculation, leading to the high complexity PA architecture.
- The possibility of the mismatching between two branches.
- The combiner can dissipate the power, resulting to decrease the overall linearity of the outphasing architecture.

Doherty PAs

Doherty PA (DPA) [3] was invented by the American electrical engineer W. H. Doherty in early 1936. The Doherty PA significantly improved the power efficiency of

conventional PA (i.e. the class-B PA from which the Doherty PA was modified). It was widely employed in commercial radio stations in Europe and Middle East in 1940s and 1950s [19]. Early Doherty PAs (from 1936 to 1979) were operated in medium- and high-power (from 50 kW to 2 MW), low frequency and medium frequency (from 30 kHz to 3 MHz) transmitters. Since then till the present, the DPAs are used in different networks and applications such as digital radio, television broadcast, and cellular networks (both base stations and handset transmitters) [19].

Figure 2.7 depicts the conceptual diagram of a classic Doherty architecture. The most basic Doherty PA configuration includes two PAs (main and auxiliary) and two $\lambda/4$ impedance inverters (one locates after the main PA and one is at the input of the auxiliary PA to balance the phase between the two paths) [18]. The main PA is usually class-B or -AB and the auxiliary PA is class-C. It is important to design the distribution/combining networks of Doherty so that the main and auxiliary PA cooperate without loading each other.

The final RF output power of DPA is combined from the power of the two PAs. At the high input power levels (typically from the peak power down to 6 dB output power back-off (OBO)), both PAs are active and together contribute to the output power. When the input power is reduced (more than 6 dB OBO from maximum

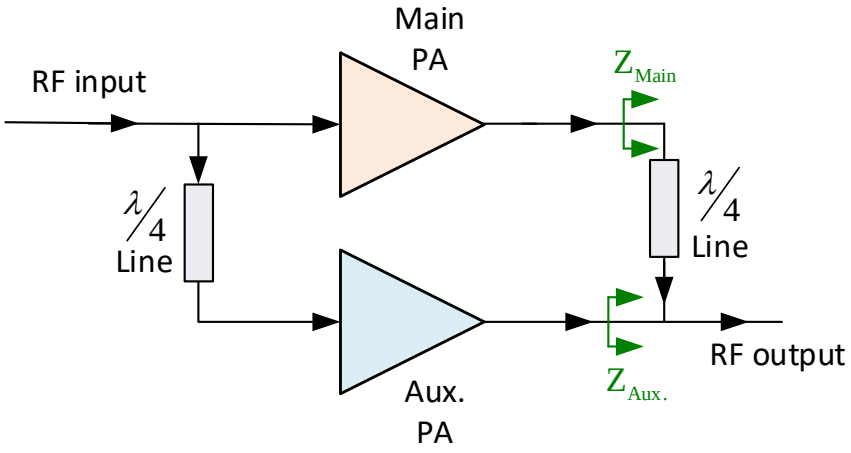


Figure 2.7: The basic Doherty amplifier configuration [18].

power), the auxiliary PA is inactivated, only the main PA operates with a modulated load impedance, thus, significantly improving the efficiency at lower power levels.

The limitations of DPAs are:

- Same as outphasing, Doherty technique requires extra RF PA(s).
- DPAs support only the narrow bandwidth signals due to the use of the quarter wavelength combining transformer. This introduces a challenge in LTE-advanced systems which work with not only wider bandwidth but also multiple-band signals [20].

2.2.2 Dynamic Supply Techniques

Envelope Tracking PAs

Conventional PAs are supplied with a fixed voltage at every instant of the transmission. The PAs can only achieve their peak efficiency at the peaks of the amplitude of the input signal and have low efficiency at low amplitude levels of the input signal. In communication systems that use high PAPR signals, this issue is only aggravated. In order to avoid the excess of the dc supply power, the envelope tracking technique [4] was proposed with the idea that the supply power is dynamically adapted to the amplitude of the PA input signal.

Figure 2.8 depicts the envelope tracking system. The amplitude detector takes the envelope of the input of PA. Then the supply modulator drives the power supply based on this input signal's envelope to provide the dynamic voltage for the PA. The PA operates in linear mode but still achieves high power efficiency due to the dynamic voltage supply.

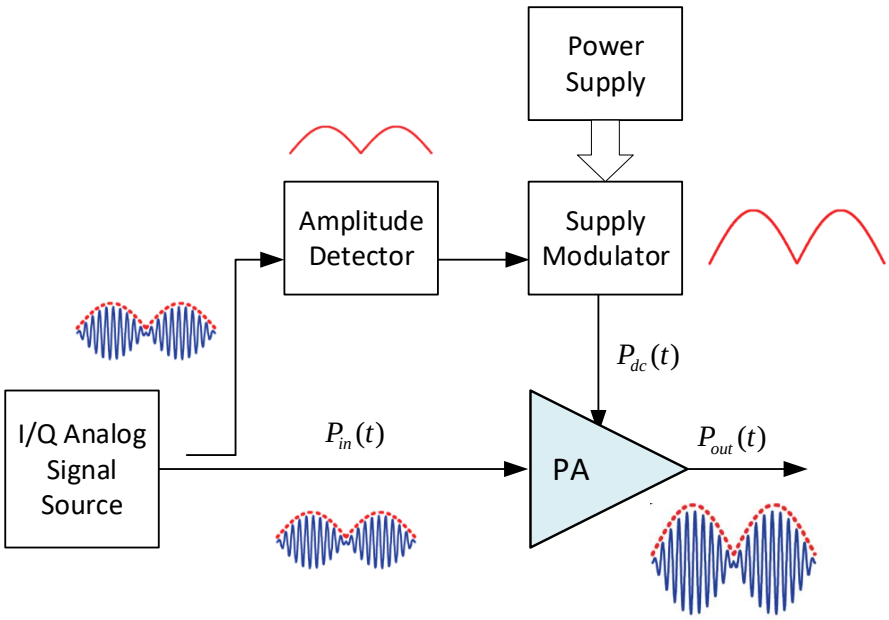


Figure 2.8: Block diagram of envelope tracking system.

In 2013, the ET technique was employed for the first time in 3G/4G LTE mobile devices by Qualcomm Inc.. Nowadays, this technique is popularly used in cellular networks in both base stations and handset transmitters [21, 22].

The ET technique's advantage is its simple design structure. The ET can be applied on conventional PAs by just simply replacing the conventional static supply for a dynamic one. However, the ET technique still has following drawbacks:

- The energy savings in the RF PA may be lost if the envelope detector and the supply modulator circuits are inefficient. Thus, the overall ET PA efficiency is a product between the efficiency of the RF PA and that of the supply modulator.

- The supplied power to the PA should have the same speed of the signal's envelope. In OFDM-based modulations the envelope bandwidth is around 3-4 times the bandwidth of the baseband complex modulated signal [23]. Therefore, some solutions are given to reduce the envelope bandwidth, such as using a slower version of the original envelope as in [23] and [11].

- The variation of the supply voltage results nonlinearity into the PA, leading to the requirement for a linearization technique (e.g. DPD) to satisfy efficiency and linearity trade-off.

Envelope Elimination and Restoration

EER was introduced in [5] by L. R. Kahn in 1952. Since then, the technique has been received lots of attention due to its high linearity and power efficiency [24–26]. Figure 2.9 shows the block diagram of EER technique.

Similar to ET technique, the envelope of the input signal is detected by the amplitude detector. Then, the supply modulator generates the supply voltage based on this envelope. However, different from ET, in ERR technique, the amplitude information of the input signal is eliminated by a limiter to gain a new signal with constant amplitude. Then the new signal is amplified by a nonlinear PA (NLPA), achieving a high efficiency. The amplitude information of the output signal of the NLPA is restored

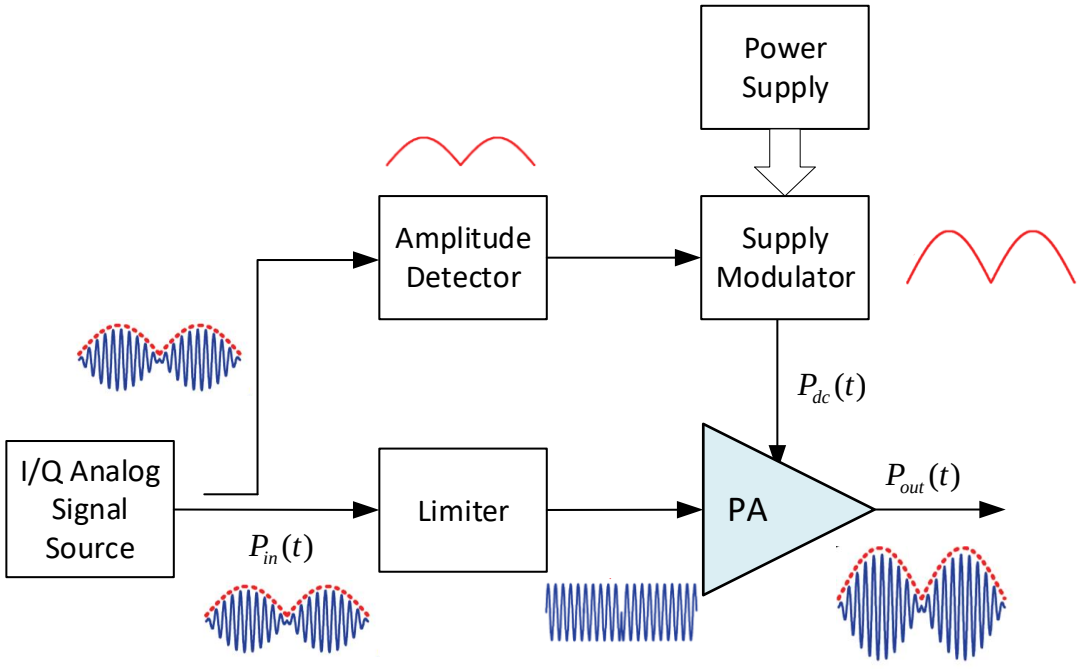


Figure 2.9: Block diagram of EER system.

by the dynamic supply voltage provided by the supply modulator. By applying the envelope tracking and constant amplitude amplifying, the EER technique enables the NLPA to work as a high efficient LPA.

The main differences between ET and EER can be summarized as following [27]:

- ET uses a linear PA (LPA) while EER uses a nonlinear PA.
- ET PA amplifies the original input signal which includes both the amplitude and phase information. Whereas, EER amplifies the signal that contains only phase information.
- The purpose of supply modulator in ET technique is to achieve the power efficiency of the LPA, while the supply modulator of EER technique aims to restore the amplitude information of the output signal of the NLPA.
- While ET can use a slow envelope [11] to reduce the bandwidth that the envelope occupies, EER requires high accurate dynamic supply. This results the requirement of high bandwidth in the components and control signals, making EER to be less attractive for wide-band OFDM-based communications.
- The synchronization between the phase and the envelope in EER technique is critical while it is not in ET.

Figure 2.10 summarizes the average power efficiency (η_{AVG}) of different high-efficiency amplification techniques (Chireix outphasing, Doherty, ET and EER), class-A, class-B PAs and ideal class-B PAs applied gate switching (GS) technique. The average power efficiency of the PAs is presented as the functions of the ratio of output power and peak output power (P_{out}/P_{outPEP}). As it can be seen from Figure 2.10, EER is the one that provides best efficiency among the considered techniques. When operating in back-off, the ET technique is more efficient than other techniques, except the EER. Whereas, when operating close to saturation, ET is slightly less efficient than EER, Chireix outphasing and Doherty. Both Chireix outphasing and Doherty

achieve high efficiency at low back-off level but become much less efficient when go deeper into the back-off. However, in general, all EET, ET, Doherty and Chireix outphasing enhance much better efficiency in comparison to the class-A PAs, class-B PAs and ideal class-B PAs applied GS technique.

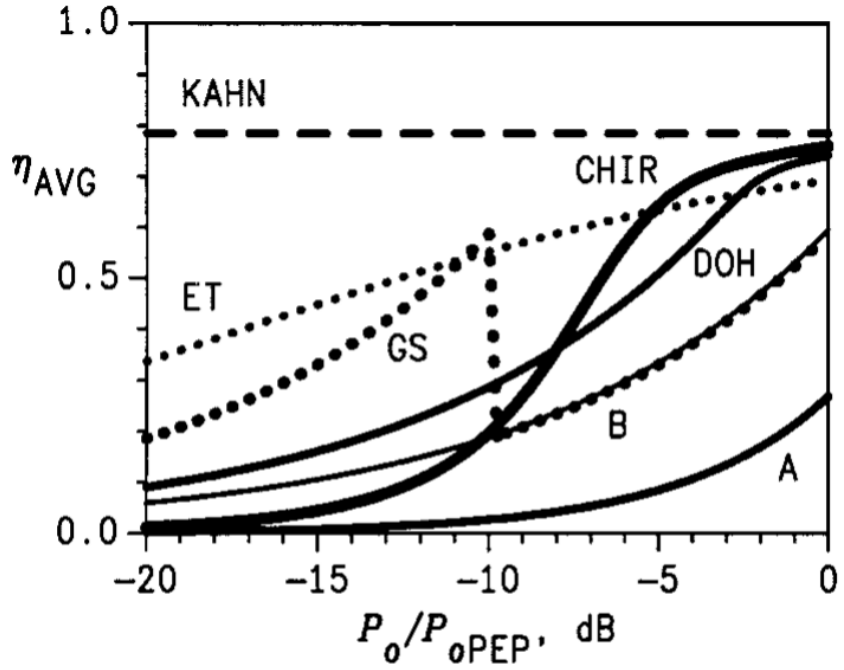


Figure 2.10: Comparison of efficiency of different high-efficiency amplification techniques and classic PAs; A: class-A PA, B: class-B PA, DOH: Doherty, GS: gate-switching technique applied for ideal class-B PA, CHIR: Chireix outphasing, ET: envelope tracking, KAHN: EER [15].

Table 2.1: Comparison of outphasing, Doherty, ET and EER techniques (a part of the Table is referenced from [27]).

	Outphasing	Doherty	ET	EER
RF PAs	2 NLPAs	LPA (carrier PA) NLPA (peak PA)	1 LPA	1 NLPA
Efficiency	Medium	Medium	High	Highest
Linearity	Low	Medium	Good	Medium/Good

Table 2.1 shows the comparison among outphasing, Doherty, ET and EER tech-

niques about the number and type of used PAs, power efficiency and linearity.

2.3 Linearization

As the demand for high data transmission rates increases, higher bandwidths (at mm-wave bands for example) and spectrally efficient modulation signals are being used. Multi-carrier techniques (e.g., OFDM-based) are aimed at maximizing spectrum efficiency, but at the price of presenting high PAPRs (i.e., larger than 10 dB, as depicted in Table 2.2). The amplification of these signals can be challenging from both the power efficiency and linearity perspective. This Section focuses on the most significant linearization techniques: feedforward, feedback and predistortion. At the end of the Section, the comparison of supported bandwidth, linearity and efficiency among the three techniques will be given.

Table 2.2: PAPR and bandwidth of the envelope signal for different wireless communication systems [27].

WCS	Signal modulation scheme	PAPR (dB)	BW (MHz)
2G	GMSK (TDMA)	0	0.2
2.5G	EDGE	3.2	0.2
3G	WCDMA	$3.5 \sim 9$	5
4G	OFDM	$8.5 \sim 13$	$2.4 \sim 20$

2.3.1 Feedforward Linearization

Feedforward was firstly introduced by H. S. Black in 1923 at Bell Telephone Laboratories [8]. Figure 2.11 shows the block diagram of feedforward linearization. The idea of the feedforward technique was stated by Black as follows: "*[..] I immediately observed that by reducing the output to the same amplitude as the input and subtracting one from the other, the distortion products only would remain which could then be amplified in a separate amplifier and used to cancel out the distortion products in*

the original amplifier output." [28].

Following the idea, the linearization system was designed consisting of two loops. First, the main loop includes the main PA that is used to amplify the input signal. The amplification of the main PA produces distortion into the output signal of the PA. Second, the auxiliary loop includes the auxiliary PA that amplifies the distortion of the main loop. Then, the output of the first loop is subtracted by the output of the second loop to produce the undistorted amplified replica of the input signal. The phase shift introduced in both loops is for the distortion cancelation adjustment [8].

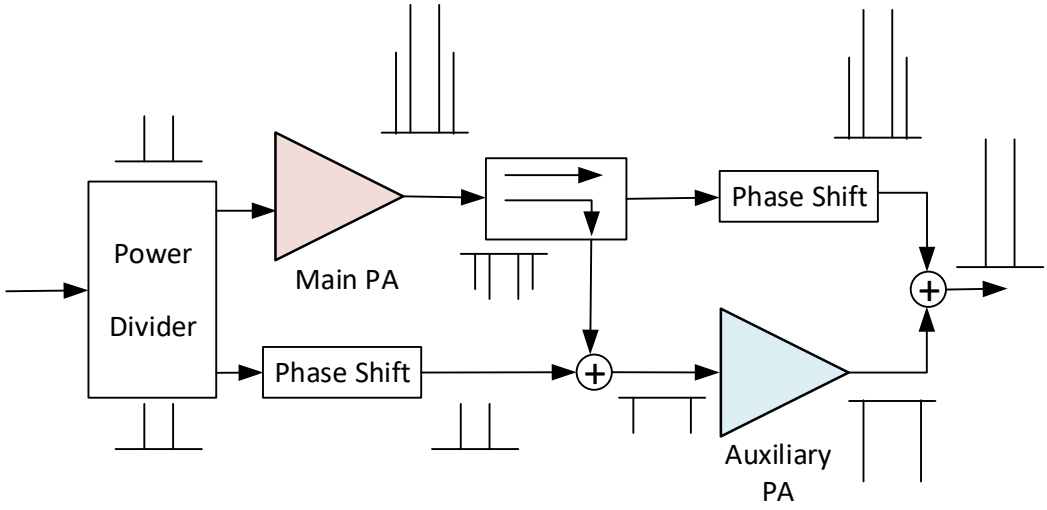


Figure 2.11: Simplified feedforward block diagram and principles of operation [29].

The advantage of the feedforward is that the concept is simple. However, some disadvantages are listed in the following.

- The achieved power efficiency is low because the auxiliary PA causes additional power consumption.
- Linearity performance is degraded because of the delay mismatch among the components.

Feedforward technique was not applied for commercial applications at the time it was invented since the technique was inappropriate to the popularly used valve

amplifiers at that time. However, after the valve amplifiers were replaced by the solid state amplifiers in the 1960s and 1970s, feedforward was used to address the high linearity requirement in cellular telephone systems [8]. In practice, distortion cancellation can be achieved only about 6 dB from saturation [8].

2.3.2 Feedback Linearization

After inventing the feedforward linearization system, in 1927, H. S. Black introduced the negative feedback principle. The negative feedback is considered the most significant breakthrough of the field of electronics in the twentieth century. The technique has been used not only in telecommunications, but also in industrial, military, and consumer electronics, weaponry, analog computers, and such biomechanical devices as pacemakers [30].

The main idea of using negative feedback for linearization purposes is feeding back the output (after phase inverting) to the input to reduce the distortion at the PA output. The block diagram of feedback linearization is shown in Figure 2.12. The feedback technique is categorized into four groups: RF feedback, envelope feedback, polar feedback and Cartesian feedback [29]. The details of the four groups can be found in [13].

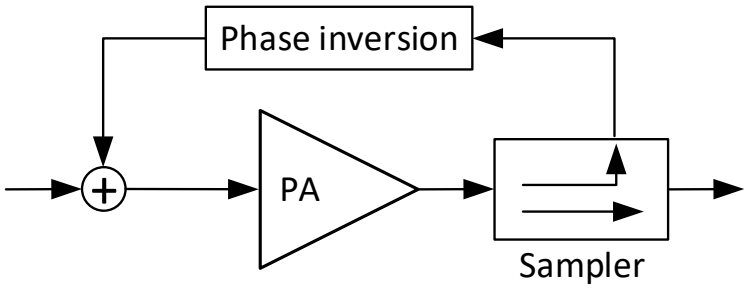


Figure 2.12: General block diagram of feedback linearization technique [29].

The feedback technique has better performance in comparison with the feedforward since it does not require additional PA and creates low distortion [28]. The

limitation of the feedback is that it can only operate with low bandwidth signal due to the loop delay (Δt_s). The bandwidth of the operated signal should be less than a quarter of Δt_s [8].

2.3.3 Predistortion

Predistortion was introduced by Adel A. M. Saleh and his colleague J. Salz from Nokia Bell Labs in 1983 [31]. Thanks to appearance of digital signal processors it became the preferred linearization technique due to its simplicity, computational efficiency, and ability for supporting for wide-band signals at high frequencies [8]. In the last three decades, many publications have been contributing to this linearization method. Some of these can be found in [10, 11, 32–36].

The predistortion linearization includes a predistorter that is located before the PA. The objective of the predistorter is to compensate for the nonlinear behavior of the PA. To do so, the predistorter creates an inverse PA nonlinear behavior (i.e., $F(.)$) on the input signal $\boldsymbol{u}$. The signal after predistorting is $\boldsymbol{x} = F(\boldsymbol{u})$. Then, the predistorted signal ($\boldsymbol{x}$) is sent to the nonlinear PA (i.e., $G(.)$) to be amplified. Consequently, the resulting output $\boldsymbol{y}$ is linear to the input signal $\boldsymbol{u}$: $\boldsymbol{y} = G(\boldsymbol{x}) = G(F(\boldsymbol{u}))$. The basic operating principle of predistortion linearization technique is illustrated in Figure 2.13.

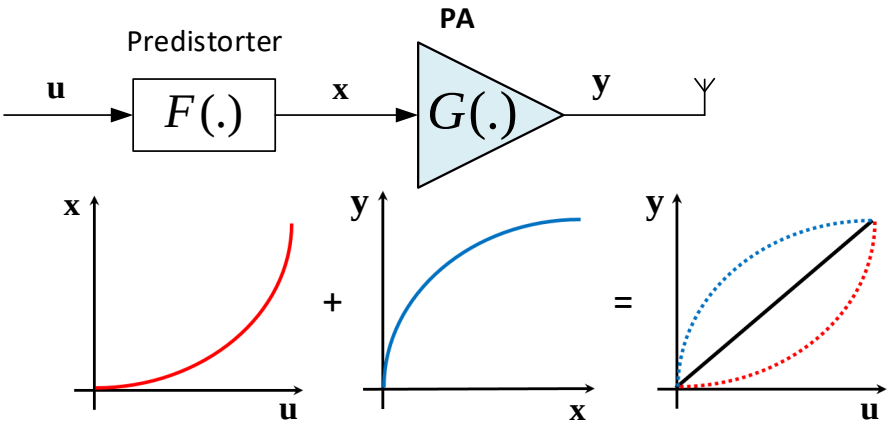


Figure 2.13: Principle of predistortion linearization technique.

Predistortion can be classified based on:

- technology: analog predistortion (APD) and digital predistortion.
- frequency band: baseband (BB), intermediate frequency (IF) and radio frequency (RF) predistortion.
- type of loop: closed-loop (or adaptive) and open-loop (or non-adaptive) predistortion.

APD performs the PA linearization in the RF domain. APD is suitable for applications that require large bandwidth (300 GHz – 300 MHz), for example satellite communications, since its operating cost is less than digital predistortion [8]. Whereas, DPD predistorts baseband signal using a digital engine. DPD is popularly used in telecommunication applications.

BB and IF predistortions have some advantages over RF predistortion. First, BB and IF are more robust with environmental coefficients (e.g. PA's temperature). Second, they do not depend on the frequency band of operation. Third, the cost of analog to digital converters (ADCs) and digital to analog converters (DACs) is lower when operating at BB and IF frequency [13]. However, the drawback of BB and IF is that the up-converters can introduce further distortion.

Closed-loop and open-loop predistortion will be discussed further in Subsection 3.4.2. This dissertation focuses on the RF PA linearization using baseband adaptive digital predistorters.

Table 2.3: Comparison of PA Linearization Techniques [37].

Technique	Bandwidth	Linearity	Efficiency
Feedforward	High	High	Low
Feedback	Low	Low	Medium
APD	Medium/High	Medium	High

Table 2.3 summarizes a comparison among linearization techniques. It can be seen

from Table 2.3 that among the three linearization techniques, predistortion is the one that enables high power efficiency, supports wide bandwidth signals and achieves good linearity levels.

Chapter 3

Principles of Digital Predistortion Linearization

As mentioned in Subsection 2.3.3, the DPD function is oriented to generate the inverse PA's nonlinear behavior. Therefore, in order to build a DPD function, first of all, the PA nonlinear behavior should be characterized via PA behavior modeling (or black-box modeling). PA behavioral modeling is a process to identify the nonlinear characteristics and memory effects of PAs with parametric mathematical models. The PA behavioral models used to form the DPD function is presented in Section 3.1. This Section also describes the procedure for converting a polynomial-based DPD model into a LUT-based model targeting FPGA implementation in Subsection 3.1.3. Generally, the coefficients of these models are found by means of the LS regression. Section 3.2 presents the identification of PA behavioral models, while Section 3.3 introduces popular numerical methods to address the LS solution. Finally, in Section 3.4, two architectures to design a closed loop adaptive DPD system: the direct and indirect learning approaches are described.

3.1 Power Amplifier Behavioral Modeling

PA behavioral models are mathematical descriptors of the PA nonlinear behavior and its memory effects. The construction of PA behavioral models are built upon a set of

PA input-output observations. Consequently, the accuracy of the model mainly relies on the adopted model structure and coefficient extraction method. Typically, the model used to approximate the PA behavior is also employed to estimate its inverse behavior. Therefore, the behavioral models presented in this Section are valid for both the PA behavior approximation and DPD linearizer description.

It is possible to find in literature an enormous amount of publications on PA behavioral modeling. The PA behavioral models can be divided into four types: single-input single-output (SISO) [38, 39], multi-input single-output (MISO) [40], single-input multi-output (SIMO), multi-input multi-output (MIMO) [40–42] systems. In this dissertation, SISO and MISO models have been considered for DPD linearization purposes. Therefore, this Section focuses on SISO and MISO models using both polynomial-based and LUT-based implementation.

3.1.1 SISO Behavioral Models

Volterra series

Volterra series was introduced by V. Volterra in 1930 [39], and is aimed at describing time-invariant nonlinear systems with fading memory. The discrete-time low-pass equivalent Volterra series formulation is described in the following. Considering the general input-output notation in Figure 3.1, the estimated output $\hat{y}[n]$ of the discrete-time low-pass equivalent Volterra series is

$$\hat{y}[n] = \sum_{p=1}^{P} \sum_{q_p=0}^{Q_p-1} \cdots \sum_{q_1=0}^{Q_1-1} h_p(q_1, \cdots, q_p) \prod_{i=1}^{p} x[n-q_i] \tag{3.1}$$

with P being the number of kernels of the series (the polynomial degree or the order of the nonlinearity), h_p the coefficient associated with the $p_t h$ kernel, $Q_1 ... Q_p$ the memory depths (or number of delays) in each kernel and n denotes the discrete time.

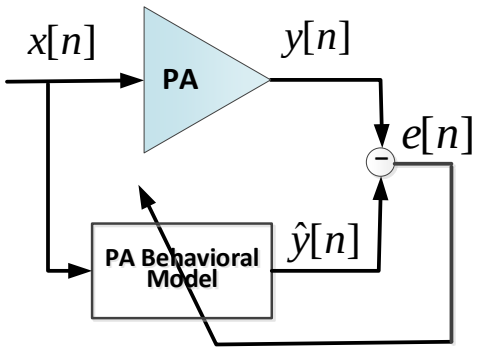


Figure 3.1: Identification of the power amplifier behavior.

The Volterra series has a simple structure. It is a linear combination of nonlinear basis functions, in which the contribution of each basis function into the model is controlled by its coefficients. The coefficient extraction of the model can be easily performed by employing linear regression techniques, such as the LS regression.

However, the main problem of the Volterra series is that the number of coefficients of the model grows significantly when the polynomial degree and memory depth increases. The high order model may include unnecessary information and lead to ill-conditioned problems in the coefficient extraction when using LS. A common solution for this is using pruning techniques. The pruning techniques apply some reduction method to the general Volterra series to get rid of irrelevant information of the series and retain only the most important terms. Some pruned Volterra series can be found in [43–45].

Another method for simplifying Volterra series expansion is the modular approach that includes the combination of components from the memoryless (or static) nonlinear and linear time-invariant dynamic subsystems (i.e., Wiener and Hammerstein models [46]). One of the most widely used models in literature due to its simplicity is the memory polynomial (MP), presented in [47]. Another common model for SISO systems is the generalized memory polynomial (GMP) behavioral model, proposed in [48]. Besides, there are plenty of other behavioral models in literature used for DPD purposes in SISO systems, just to mention a couple of examples, the NARMA

model proposed in [49] and the Dynamic Deviation Reduction Volterra series in [44]. Further information on PA behavioral models for SISO systems can be found in [38].

Memory Polynomial

One of the simplest PA behavioral models capable to characterize both nonlinear distortion and memory effects is the memory polynomial, presented in [47]. In the MP the estimated output $\hat{y}[n]$ is defined as follows

$$\widehat{y}[n] = \sum_{i=0}^{L-1} \sum_{p=0}^{P-1} \alpha_{pi}\ x[n-\tau_i]\ \left|x[n-\tau_i]\right|^p \tag{3.2}$$

where P is the nonlinearity order of the polynomial, L the length of the memory, α_{pi} the complex coefficients describing the model, and τ (with $\tau \in \mathbb{Z}$ and $\tau_0 = 0$) the delay shifts (i.e. the most important non-consecutive delays of the input signal $x[n]$ that better contribute to characterize the PA memory effects). The total number of coefficients of the MP model is in the order of $M = P \cdot L$.

Generalized Memory Polynomial

The generalized memory polynomial behavioral model, proposed in [48], is another widely used model for SISO systems. GMP is defined as following.

$$\begin{aligned}\widehat{y}[n] = &\sum_{i=0}^{L_a-1} \sum_{p=0}^{P_a-1} \alpha_{pi}\ x[n-\tau_i^a]\ \left|x[n-\tau_i^a]\right|^p + \\ &\sum_{j=1}^{K_b} \sum_{i=0}^{L_b-1} \sum_{p=1}^{P_b} \beta_{pij}\ x[n-\tau_i^b]\ \left|x[n-\tau_i^b-\tau_j^b]\right|^p + \\ &\sum_{j=1}^{K_c} \sum_{i=0}^{L_c-1} \sum_{p=1}^{P_c} \gamma_{pij}\ x[n-\tau_i^c]\ \left|x[n-\tau_i^c+\tau_j^c]\right|^p\end{aligned} \tag{3.3}$$

where P_a, P_b, P_c are the nonlinearity orders of the polynomials, L_a, L_b, L_c, K_b, K_c are the lengths of memories. α_{pi}, β_{pij} and γ_{pij} are the complex coefficients describing

the model, and τ^a, τ^b and τ^c (with $\tau \in \mathbb{Z}$ and $\tau_0 = 0$) are the most significant non-consecutive delays of the input signal $x[n]$ that better contribute to characterize memory effects. The total number of coefficients of GMP model is $M = P_a L_a + P_b L_b K_b + P_c L_c K_c$.

Unlike the MP, the GMP has bi-dimensional kernels (considering cross-term products between the complex signal and the lagging and leading envelope terms) which increases the accuracy of the modeling at the price of increasing the number of coefficients.

3.1.2 MISO Behavioral Model

When considering concurrent multi-band transmissions such as in [50], or even combined with PA dynamic supply modulation strategies such as in [51], or also in multi-antenna systems where each transmit path has its own PA and antenna element such as in [52]; MISO behavioral models are required to characterize the different sources of nonlinear behavior. In concrete, this Subsection presents the 3-D distributed memory polynomial (3D-DMP) model introduced in [11] to approximate the behavior of a concurrent dual-band envelope tracking PA.

3-D Distributed Memory Polynomial

The MISO behavioral model 3D-DMP has a parallel structure including three branches, in which each branch is responsible for characterizing/compensating one of the three main unwanted nonlinear distortion effects (intra-band, cross-band and dynamic supply distortion) in concurrent dual-band envelope tracking PAs. Figure 3.2 depicts the block diagram of the model.

The 3D-DMP behavioral model (for Band 1 signal) is defined as

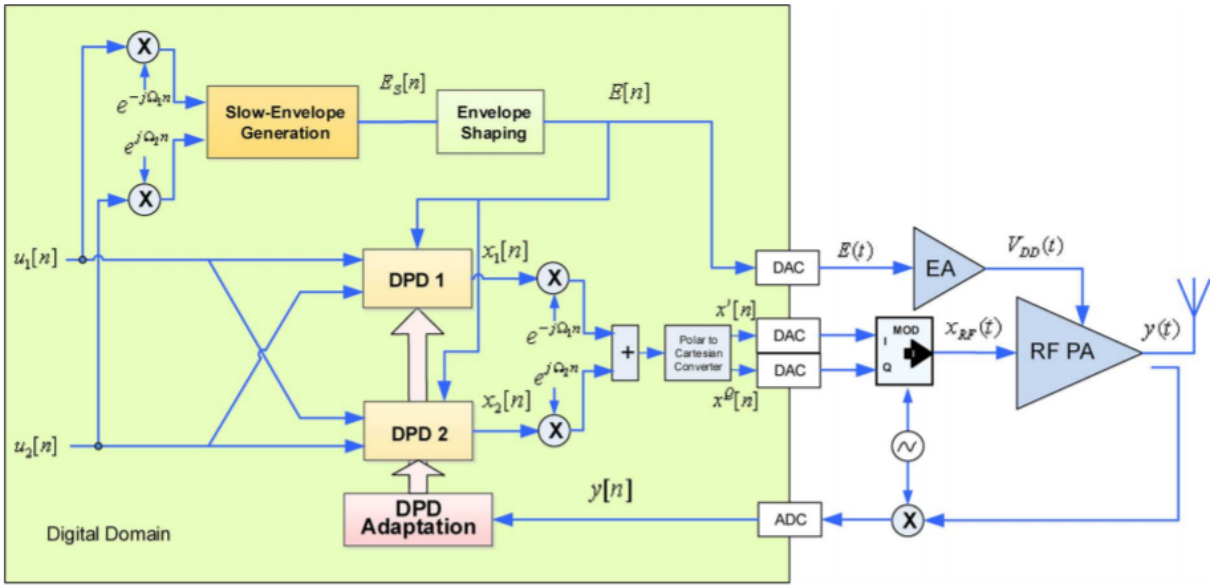


Figure 3.2: Block diagram of the 3D-DPD and slow envelope generator for concurrent dual-band ET PAs [11].

$$\begin{aligned}\widehat{y}[n] &= \sum_{i=0}^{N_1-1}\sum_{p=0}^{P_1-1} a_{pi}.x_1[n-\tau_i^{x_1}]|x_1[n-\tau_i^{x_1}]|^p \\ &+ \sum_{i=1}^{N_2-1}\sum_{p=0}^{P_2-1} \beta_{pi}(x_2).x_1[n]|x_1[n-\tau_i^{x_1}]|^p \\ &+ \sum_{i=1}^{N_3-1}\sum_{p=0}^{P_3-1} \gamma_{pi}(E).x_1[n]|x_1[n-\tau_i^{x_1}]|^p\end{aligned} \tag{3.4}$$

where

$$\beta_{pi}(x_2) = \sum_{j=1}^{M_2-1}\sum_{q=0}^{Q_2-1} b_{piqj}|x_2[n-\tau_j^{x_2}]|^q \tag{3.5}$$

and

$$\gamma_{pi}(E) = \sum_{k=1}^{K_3-1}\sum_{r=0}^{R_3-1} c_{pirk}(E[n-\tau_k^e]^r \tag{3.6}$$

P_1, P_2 and P_3 are the polynomial orders of the Band 1 signal x_1 at each branch. N_1, N_2 and N_3 are the number of delays of the signal x_1 at each branch. Q_2 and M_2 are the polynomial order and the number of delays of the interference signal, respectively. K_3 and R_3 are the polynomial order and the number of delays of the supply envelope E, respectively. τ^{x_1}, τ^{x_2} and τ_k^e are the most significant sparse delays of the input signal x_1, the interference signal x_2 and the supply envelope E, respectively. a_{pi}, b_{piqj} and c_{pirk} are the coefficients of the model. The 3D-DMP behavioral model for Band 2 signal can be defined similarly to the Band 1 signal, considering the input signal x_2 and interference signal x_1.

The 3D-DMP model will be used in Section 6.2 to compare the model order reduction capabilities vs. linearization performance when considering two different dimensionality reduction techniques.

3.1.3 Look-up table implementation of DPD models

FPGA is an integrated circuit containing an array of programmable logic blocks that is reconfigurable and reprogrammable to allow flexible computing as performed in computer software. FPGA is an attractive solution for implementing the DPD function. Some of the advantages of fast prototyping DPD in FPGA platforms are the high-speed processing, high density integration, flexible implementation and parallel operation mechanisms [53]. In general, the DPD function can be implemented in FPGA either by the polynomial based or the look-up table (LUT) based methods [54]. The direct implementation of polynomial-based DPD models in FPGA requires several complex multiplications and additions [55]. Whereas, LUT based implementation reduces the FPGA logic resources for describing the nonlinear function. The authors in [56] discussed the advantages of LUT-based to polynomial-based implementation of a DPD model. First, the complexity of LUTs is lower than polynomials in terms of using fewer multipliers (which is one of the most expensive elements in the FPGA hardware). Reducing the number of multipliers helps to reduce hardware resources

requirements for the DPD implementation. Second, LUTs are more numerically stable than polynomials when the order of the polynomials is high. According to the rule of thumb, a 32-bit processor cannot handle polynomials greater than fifth or sixth order. With higher order polynomials, the numerical instability becomes apparent. Third, in the case that the signal power is unstable, a polynomial-based predistorter is less robust than a LUT-based one.

In this dissertation, both polynomial based and LUT-based behavioral models are considered to validate the suggested dimensionality reduction algorithms for the DPD coefficient identification. The LUT-based models used in this dissertation follow the linear interpolation and extrapolation approach described in [57, 58]. For example, the GMP SISO behavioral model in (3.3) can be implemented with 1-D LUTs [58], as it will be shown in Section 5.4 of this dissertation. Whereas the 3D-DMP MISO behavioral model in (3.4) can be mapped into LUTs using 1-D LUTs and 2-D LUTs [57] as it will be shown in Section 6.2.

The 1-D LUT [58] is a piecewise linear complex function, defined in (3.7) as the linear combination of K basis functions.

$$f_{\overline{\Phi}}(u) = \sum_{i=0}^{K-1} \varphi_i \Lambda_{g(i,K)}(u - i\delta) \tag{3.7}$$

where u is a real number, $g(i,K) = \begin{cases} 0, & i < K-2 \\ 1, & i = K-2 \\ 2, & i = K-1 \end{cases}$; $\delta = max(u)/(K-1)$ is the width of each region on the real interval at which function $f_{\overline{\Phi}}(u)$ is defined; $\Lambda_0(u)$ defined in (3.8) is the interpolation basis function on the interval $[0, (K-1)\delta]$; while $\Lambda_1(u)$ in (3.9) and $\Lambda_2(u)$ in (3.10) are extrapolation basis functions on the interval

$[(K-1)\delta, \infty]$. Finally, φ_i are the coefficients of the picewise complex function.

$$\Lambda_0(u) = \Big(1 - \frac{|u|}{\delta}\Big) w\Big(\frac{|u|}{\delta}\Big); w(u) = \begin{cases} 1, \ 0 \leq u \leq 1 \\ 0, \ otherwise \end{cases} \tag{3.8}$$

$$\Lambda_1(u) = \Big(1 - \frac{|u|}{\delta}\Big) s\Big(u + \delta\Big); \ s(u) = \begin{cases} 1, \ u \geq 0 \\ 0, \ otherwise \end{cases} \tag{3.9}$$

$$\Lambda_2(u) = \Big(1 + \frac{u}{\delta}\Big) s\Big(u + \delta\Big); \ s(u) = \begin{cases} 1, \ u \geq 0 \\ 0, \ otherwise \end{cases} \tag{3.10}$$

The 2-D LUT [57] is defined by a piecewise bilinear complex function as follows,

$$f_{\overline{\Phi}}(u_1, u_2) = \sum_{i=0}^{K_1-1} \sum_{j=0}^{K_2-1} \varphi_{i,j} \Gamma_{g(i,K_1),g(j,K_2)}(u_1 - i\delta_1, u_2 - j\delta_2) \tag{3.11}$$

where $\Gamma_{i,j}(u_1, u_2) = \Lambda_i(u_1)\Lambda_j(u_2)$. Functions $g(i, K)$, $\Lambda_0(u)$, $\Lambda_1(u)$, $\Lambda_2(u)$, $w(u)$ and $s(u)$ are defined in (3.7)-(3.10). In (3.11), u_1 and u_2 are real numbers; K_1 and K_2 are the numbers of basis functions in the u_1 and u_2 directions; δ_1 and δ_2 are the widths of each region of u_1 and u_2 respectively, $\delta_1 = max(u_1)/(K_1 - 1)$ and $\delta_2 = max(u_2)/(K_2 - 1)$. Further details on the bilinear interpolation and extrapolation can be found in [57].

3.2 Identification of PA Behavioral Models

In general, the estimated PA behavioral model output $\widehat{y}[n]$ (for $n = 0, 1, \cdots, N-1$), can be defined in a matrix notations as

$$\widehat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{w} \tag{3.12}$$

where $\boldsymbol{w} = \Big(w_1, \cdots, w_i, \cdots, w_M\Big)^T$ is the $M \times 1$ vector of coefficients. $\boldsymbol{X}$ is the $N \times M$ data matrix (with $N \gg M$) containing the basis functions of the PA behavioral model. N is the number of samples of the amplified signal. M is the number of columns (i.e. the basis functions) of $\boldsymbol{X}$. M is so-called the order of the behavioral model.

The data matrix $\boldsymbol{X}$ can be defined as

$$\boldsymbol{X} = \Big(\boldsymbol{\varphi}_x[0], \boldsymbol{\varphi}_x[1], \cdots, \boldsymbol{\varphi}_x[n], \cdots, \boldsymbol{\varphi}_x[N-1]\Big)^T \tag{3.13}$$

where $\boldsymbol{\varphi}_x[n] = \Big(\phi_1^x[n], \cdots, \phi_i^x[n], \cdots, \phi_M^x[n]\Big)^T$ is the $M \times 1$ vector of basis functions $\phi_i^x[n]$ (with $i = 1, \cdots M$) at time n.

This general equation can be particularized for any behavioral model. Thus, for example, taking into account the MP model in (3.2), the basis functions in (3.13) can be defined as

$$\boldsymbol{\varphi}_x[n] = \Big(x[n], \cdots, x[n-\tau_i]\ \big|x[n-\tau_i]\big|^p, \cdots, x[n-\tau_{L-1}]\ \big|x[n-\tau_{L-1}]\big|^{P-1}\Big)^T \tag{3.14}$$

Similarly, the original coefficients of the MP, α_{pi}, are mapped into w_i coefficients, with $i = 1, \cdots, M$. In order to compute the estimated output signal $\widehat{\boldsymbol{y}}$, the vector of coefficients $\boldsymbol{w}$ has to be found.

Generally, the problem in (3.12) has no exact solution since it is *over-determined* (i.e. more equations than unknowns). To identify the vector of coefficients $\boldsymbol{w}$ we define a cost function that takes into account the identification error $\boldsymbol{e}$ expressed, as

depicted in Figure 3.1, as

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y} - \widehat{\boldsymbol{y}}. \tag{3.15}$$

Taking the ℓ_2-norm squared of the identification error, the least squares minimization problem can be defined as follows

$$\begin{aligned} &\text{Given } \boldsymbol{X} \in \mathbb{C}^{N\times M}, N \gg M, \boldsymbol{y} \in \mathbb{C}^N, \\ &\text{find } \boldsymbol{w} \in \mathbb{C}^M \text{ such that } ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 \text{ is minimized.} \end{aligned} \tag{3.16}$$

It can be proven that the solution to the LS problem in (3.16) is given by

$$\boldsymbol{w} = (\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{y} \tag{3.17}$$

with H denotes Hermitian transpose. The solutions to the LS problem will be further discussed in Section 3.3.

The accuracy of the PA behavioral model depends on the adopted model structure and the coefficient extraction procedure (usually solved via LS). In order to evaluate the accuracy of a PA behavioral model, the measurements *normalized mean squared error* ($NMSE$) and *adjacent channel error power ratio* ($ACEPR$) are used. The $NMSE$ is used to measure the resemblance of the estimated output $\widehat{y}(n)$ and measured output signal $y(n)$ of the PA. It is defined as

$$NMSE(\text{dB}) = 10\log_{10}\left(\frac{\sum\limits_n |y(n) - \widehat{y}(n)|^2}{\sum\limits_n |y(n)|^2}\right) \tag{3.18}$$

And the $ACEPR$ measures the power of the error signal in the adjacent channels relative to the power inside the channel. It is given by

$$ACEPR(\text{dB}) = 10\log_{10}\left(\frac{\int_{adj} |E(f)|^2 df}{\int_{ch} |Y(f)|^2 df}\right) \tag{3.19}$$

where $Y(f)$ and $E(f)$ are the Fourier transforms of $y(n)$ and $e(n) = y(n) - \widehat{y}(n)$, respectively.

In following Section, some popular numerical methods used to solve least squares are presented.

3.3 Numerical Methods for Solving Least Squares

The LS problem in (3.16) can be solved via many methods, such as normal equations, QR factorization and singular value decomposition (SVD). Among of them, QR factorization approach for solving the LS (QR-LS) of DPD coefficient estimation is most numerically stable thus being adopted as the baseline method to compare with other approaches presented in this dissertation.

3.3.1 Normal Equations

This method converts the problem in (3.16) to the normal equations as follows

$$\boldsymbol{X}^H \boldsymbol{X} \boldsymbol{w} = \boldsymbol{X}^H \boldsymbol{y} \tag{3.20}$$

Then, applying Cholesky factorization [59] to the correlation matrix $\boldsymbol{X}^H \boldsymbol{X}$:

$$\boldsymbol{X}^H \boldsymbol{X} = \boldsymbol{R}^H \boldsymbol{R} \tag{3.21}$$

where $\boldsymbol{R}$ is upper-triangular matrix. In next step, what we need to do is solving two following upper-triangular systems

$$\boldsymbol{R}^H \boldsymbol{z} = \boldsymbol{X}^H \boldsymbol{y}, \text{ for } \boldsymbol{z}, \tag{3.22}$$

$$\boldsymbol{R}^H \boldsymbol{w} = \boldsymbol{z}, \text{ for } \boldsymbol{w}. \tag{3.23}$$

The operational cost of the normal equations method is $\sim NM^2 + \frac{1}{3}M^3$ flops [59]. The method is not computational expensive in comparison to the next two solutions. However, its biggest issue is that is unstable with rounding errors when finite precision arithmetic is used to implement the method.

3.3.2 QR factorization

The most commonly used approach to solve the LS problem is QR factorization. In this method, the matrix $\boldsymbol{X}$ is decomposed as

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R} \tag{3.24}$$

where $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_M]$ is an $N \times M$ unitary matrix (i.e., $\boldsymbol{Q}^H\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^H = \boldsymbol{I}$) and $\boldsymbol{R}$ is an $M \times M$ upper-triangular matrix where the diagonal entries r_{jj} are nonzero. Therefore, it can be said that each column of $\boldsymbol{X}$ is a linear combination of columns in $\boldsymbol{Q}$, that can be expressed as follows

$$\begin{aligned} \boldsymbol{x}_1 &= r_{11}\boldsymbol{q}_1, \\ \boldsymbol{x}_2 &= r_{12}\boldsymbol{q}_1 + r_{22}\boldsymbol{q}_2, \\ &\cdots \\ \boldsymbol{x}_M &= r_{1M}\boldsymbol{q}_1 + r_{2M}\boldsymbol{q}_2 + \cdots + r_{MM}\boldsymbol{q}_M. \end{aligned} \tag{3.25}$$

From (3.24), the problem in (3.16) takes the form

$$\boldsymbol{Q}\boldsymbol{R}\boldsymbol{w} = \boldsymbol{y}, \tag{3.26}$$

left-multiplication by $\boldsymbol{Q}^H$ gives

$$\boldsymbol{R}\boldsymbol{w} = \boldsymbol{Q}^H\boldsymbol{y}. \tag{3.27}$$

Solving the upper-triangular system (3.27) for $\boldsymbol{w}$ by back substitution, we got the result.

Several algorithms can be used for computing the QR factorization (3.24), the Gram-Schmidt algorithm, Household triangularization, Givens transformation method, Pivoting and Gaussian elimination [59].

3.3.3 Singular Value Decomposition

When the matrix $\boldsymbol{X}$ is not full rank, the QR factorization method becomes less stable. The alternative for solving the LS problem is the singular value decomposition (SVD) method, consisting in a matrix factorization technique (which can be real or complex). The decomposition of the matrix $\boldsymbol{X}$ through SVD factorization can be written as

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^H \tag{3.28}$$

in which, each column of the $M \times M$ matrix $\boldsymbol{V}$ (named *right singular vector*) is the unit vector of the unit sphere $\boldsymbol{S}$; each column of the $N \times M$ matrix $\boldsymbol{U}$ (named *left singular vector*) is the unit vector oriented in the direction of a principle semiaxes of $\boldsymbol{XS}$. Therefore, the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary; Σ is an $M \times M$ diagonal matrix with positive entries. In case of full SVD, $\boldsymbol{U}$ is $N \times N$, $\boldsymbol{V}$ is $M \times M$ and Σ is $N \times M$.

Then, the problem in (3.16) takes the form

$$\boldsymbol{U}\Sigma\boldsymbol{V}^H\boldsymbol{w} = \boldsymbol{y}. \tag{3.29}$$

Left-multiplication by $\boldsymbol{U}^H$ results in

$$\Sigma\boldsymbol{V}^H\boldsymbol{w} = \boldsymbol{U}^H\boldsymbol{y}. \tag{3.30}$$

Set $\boldsymbol{z} = \boldsymbol{V}^H \boldsymbol{w}$, then (3.30) turns to

$$\Sigma \boldsymbol{z} = \boldsymbol{U}^H \boldsymbol{y}. \tag{3.31}$$

Solve the diagonal system in (3.31) for $\boldsymbol{z}$, then set $\boldsymbol{w} = \boldsymbol{V}\boldsymbol{z}$.

The SVD can be solved via eigenvalue decomposition, Golub-Kahan bidiagonalization or Lawson–Hanson–Chan (LHC) bidiagonalization [59].

3.4 Adaptive Digital Predistortion Linearization

Since the inverse nonlinear behavior of a PA is built using a limited number of measured input and output samples, the coefficients of the DPD function may not be the optimum ones. Moreover, the characteristics of PA can change over time due to the environment temperature, component aging, component drift and power supply variations [60], thus the DPD coefficients should be adaptively updated. By iteratively adjusting the predistorter coefficients to minimize the nonlinear residual between the transmitted and received signal, it is possible to have a better estimation of the optimum DPD coefficients.

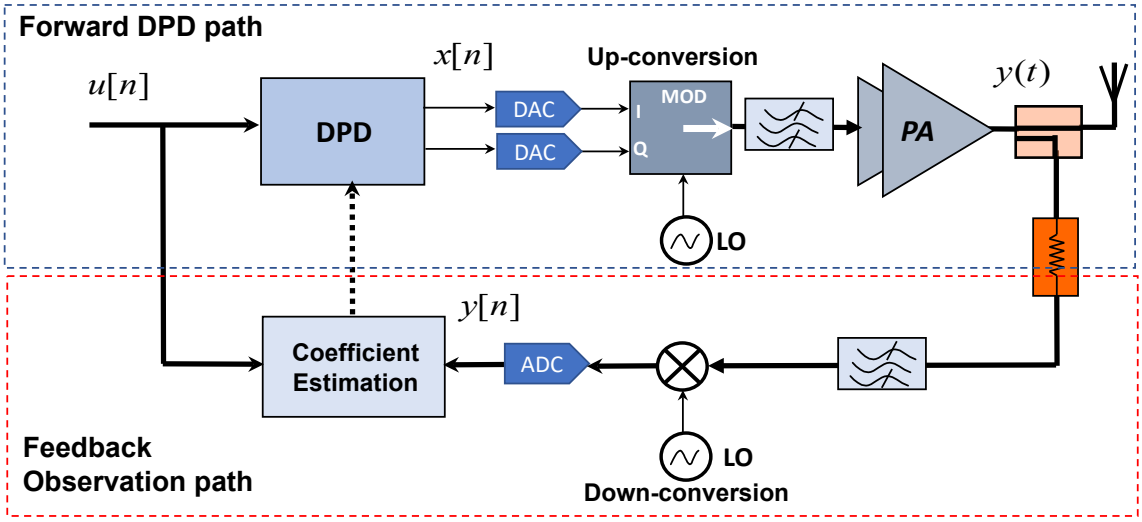


Figure 3.3: Adaptive digital predistortion linearization system.

An adaptive digital predistortion linearization system consists of two subsystems: the forward path and the feedback (or observation) path (see Figure 3.3). In the forward path, the DPD function has to operate real-time and is responsible for predistorting the input signal by generating the inverse nonlinear characteristic of the PA to be linearized. In the feedback path, the coefficients of the DPD function are estimated and updated iteratively.

3.4.1 DPD Forward Path

In the forward path, the input-output relationship at the DPD block can be described as

$$x[n] = u[n] - d[n] \tag{3.32}$$

in which $u[n]$ is the input signal of the DPD block, $x[n]$ is the output signal, and $d[n]$ is the distortion signal that can be described using the aforementioned PA behavioral models that can be found in the literature. In general, the distortion in (3.32) can be expressed as

$$d[n] = \boldsymbol{\varphi_u}^T[n]\boldsymbol{w}[n] \tag{3.33}$$

with $\boldsymbol{w}[n] = \Big(w_1[n], \cdots, w_i[n], \cdots, w_M[n]\Big)^T$ being a vector of coefficients at time n with dimensions $M \times 1$ (M being the order of the behavioral model). $\boldsymbol{\varphi_u}^T[n] = \Big(\phi_1^u[n], \cdots, \phi_i^u[n], \cdots, \phi_M^u[n]\Big)$ is the vector containing the basis functions $\phi_i^u[n]$ ($i = 1, \cdots, M$) at time n.

As explained in the previous Section, the same behavioral model or basis functions used for approximating the response of the PA can be also used for DPD purposes to estimate the inverse response of the PA. Therefore, for example, by particularizing $\boldsymbol{\varphi_u}^T[n]$ with the MP model described in (3.2), the basis function $\phi_i^u[n]$ is defined as

$$\phi_i^u[n] = u[n-\tau_l]\ \Big|u[n-\tau_l]\Big|^p \tag{3.34}$$

where p is the polynomial order and τ_l is the delay shift.

Now, (3.32) can be rewritten considering a matrix notation as follows

$$\boldsymbol{x} = \boldsymbol{u} - \boldsymbol{U}\boldsymbol{w} \tag{3.35}$$

where $\boldsymbol{u} = (u[0], \cdots, u[n], \cdots, u[N-1])^T$, with $n = 0, \cdots, N-1$, is the $N \times 1$ input vector. $\boldsymbol{x} = (x[0], \cdots, x[n], \cdots, x[N-1])^T$ is the $N \times 1$ predistorted vector. The $N \times M$ data matrix $\boldsymbol{U}$ is defined as

$$\boldsymbol{U} = (\boldsymbol{\varphi_u}[0], \cdots, \boldsymbol{\varphi_u}[n], \cdots, \boldsymbol{\varphi_u}[N-1])^T \tag{3.36}$$

Note that the matrix $\boldsymbol{U}$ in (3.36) contains the same basis functions than the matrix $\boldsymbol{X}$ in (3.13). However, while matrix $\boldsymbol{X}$ is used for approximating the response of the PA, the matrix $\boldsymbol{U}$ is used for estimating the inverse response of the PA.

The DPD function in the forward path described in (3.35) can be implemented to operate in real-time in a programmable logic (PL) device following different approaches, such as look-up tables as in [57, 61], complex multipliers following a polynomial approach using the Horner's rule as in [62], or some combination of complex multipliers and memory blocks as in [63].

3.4.2 DPD Feedback Path

The feedback path subsystem is where the DPD coefficients are estimated and adapted. Unlike the DPD in the forward path, the coefficient identification/adaptation in the feedback path is not required to be carried out in real time. Alternatively, the DPD coefficients can be iteratively extracted and adapted in a slower time-scale. Therefore, it can be implemented in a processing system (PS) instead of in a PL device.

There are two main methods for coefficient estimation and adaption:

- **direct learning approach**: In this approach, the predistorter is directly at-

tained via "pre-inverting" the PA behavior and the DPD coefficients are continually adjusted by comparing the PA output $y[n]$ to the input signal $u[n]$. This approach is also named "closed-loop" since the predistorter is inside the feedback loop (see Figure 3.4). Some applications of the direct learning approach can be found in [64–67].

- **indirect learning approach**: In this approach, first the PA nonlinear model is post-inversed by a postdistorter, then the coefficients of the postdistorter are used as the coefficients of the predistorter. Some applications of the indirect learning approach can be found in [32, 47].

A comparison of DPD indirect and direct learning approaches were presented in [68, 69]. According to the comparisons in [68] and [69], the direct learning model performs better than the indirect one since the PA's output signal can be noisy and thus reducing the accuracy of the postdistorter estimation. Consequently, the experimental results presented in this dissertation have been obtained considering DPD linearization based on the direct learning approach. Both learning methods are explained in the following.

Direct Learning approach

The block diagram of the adaptive DPD following direct learning approach is shown in Figure 3.4. The DPD function in the forward path is described in (3.33) or in (3.35). In the feedback path, the DPD coefficients can be updated iteratively as follows

$$\boldsymbol{w}^{j+1} = \boldsymbol{w}^{j} + \mu \boldsymbol{\Delta w} \tag{3.37}$$

with μ ($0 \leq \mu \leq 1$) being a weighting factor (or convergence factor) and $\boldsymbol{\Delta w}$ is the DPD coefficients increment. The LS estimation of $\boldsymbol{\Delta w}$ is defined in the following,

$$\boldsymbol{\Delta w} = \left(\boldsymbol{U}^{\mathbf{H}}\boldsymbol{U}\right)^{-1}\boldsymbol{U}^{\mathbf{H}}\boldsymbol{e} \tag{3.38}$$

where $\boldsymbol{U}$ is the $N \times M$ data matrix, $\boldsymbol{e}$ is the $N \times 1$ vector of the identification error (also known as the residual linearization error) and is defined as

$$\boldsymbol{e} = \frac{\boldsymbol{y}}{G_0} - \boldsymbol{u}. \tag{3.39}$$

where G_0 is the desired linear gain of the PA and $\boldsymbol{y}$ and $\boldsymbol{u}$ are the $N \times 1$ vectors of the PA output and the transmitted input, respectively.

If the correlation matrix $\boldsymbol{U}^H\boldsymbol{U}$ is ill-conditioned, it will negatively impact the LS estimation in (3.38). To avoid the uncertainty in the DPD coefficients estimation regularization techniques can be applied to both avoid the numerical ill-conditioning of the estimation and reduce the number of coefficients of the DPD function in the forward path, which ultimately impacts the baseband processing computational complexity and power consumption. This topic will be addressed in Section 4.1.1.

Indirect Learning Approach

The block diagram of the adaptive DPD following an indirect learning approach is shown in Figure 3.5. Unlike the direct learning approach, the indirect learning approach estimates the inverse PA model with postdistortion. This is based on the assumption that the coefficients of the postdistorter and the coefficients of the predis-

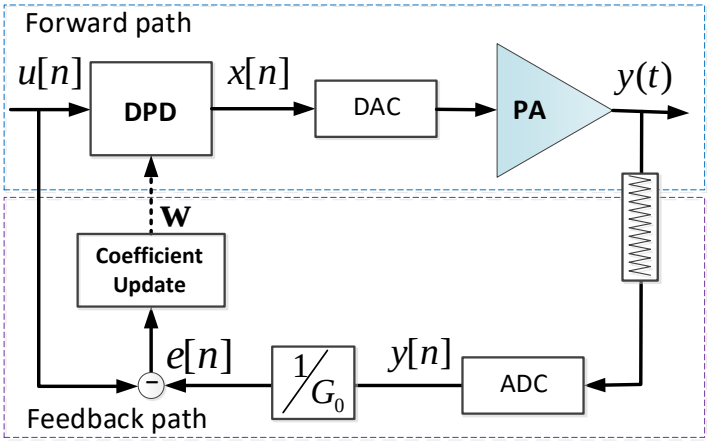


Figure 3.4: Block diagram of DPD linearization following a direct learning approach.

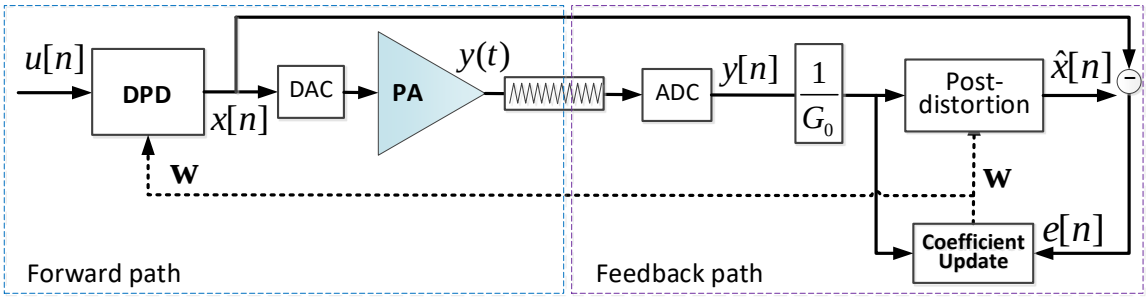


Figure 3.5: DPD linearization following indirect learning approach.

torter are equivalent. The output of postdistorter $\widehat{\boldsymbol{x}} = (\widehat{x}[0], \cdots, \widehat{x}[n], \cdots, \widehat{x}[N-1])^T$ is defined as

$$\widehat{\boldsymbol{x}} = \frac{\boldsymbol{y}}{G_0} - \boldsymbol{Y}\boldsymbol{w} \tag{3.40}$$

$\boldsymbol{y} = (y[0], \cdots, y[n], \cdots, y[N-1])^T$, with $n = 0, \cdots, N-1$, is the $N \times 1$ PA output vector (i.e. postdistorter input). The $N \times M$ data matrix $\boldsymbol{Y}$ is defined as

$$\boldsymbol{Y} = \Big(\boldsymbol{\varphi}_y[0], \boldsymbol{\varphi}_y[1], \cdots, \boldsymbol{\varphi}_y[n], \cdots, \boldsymbol{\varphi}_y[N-1]\Big)^T \tag{3.41}$$

with $\boldsymbol{\varphi}_y[n] = \Big(\phi_1^y[n], \cdots, \phi_i^y[n], \cdots, \phi_M^y[n]\Big)^T$ being the $M \times 1$ vector of basis functions $\phi_i^y[n]$ (with $i = 1, \cdots M$) at time n. At the j^{th} iteration (i.e., when considering buffers of N data samples) the new postdistorter coefficients w_j (that later will be used as predistorter coefficients) are obtained as in (3.37). However, following the indirect learning approach, $\boldsymbol{\Delta w}$ is calculated as follows

$$\boldsymbol{\Delta w} = \Big(\boldsymbol{Y}^{\mathbf{H}}\boldsymbol{Y}\Big)^{-1}\boldsymbol{Y}^{\mathbf{H}}\boldsymbol{e} \tag{3.42}$$

with $\boldsymbol{e} = (e[0], \cdots, e[n], \cdots, e[N-1])^T$ being the $N \times 1$ vector of the postdistortion estimation error, defined as

$$\boldsymbol{e} = \widehat{\boldsymbol{x}} - \boldsymbol{x}. \tag{3.43}$$

Once the postdistortion coefficients $\boldsymbol{w}$ are estimated, these are used as the coefficients of the predistorter in the forward path. The simplicity of the indirect learning

approach makes it widely used. However, there are some issues that should be taken into account. As explained in [70], the input of the postdistortion $y[n]$ can be noisy, making the postdistorter coefficients' estimation converge to biased values. In addition, since the commutative property does not apply in nonlinear systems, there is no guarantee that the postdistortion coefficients will perform well as predistortion coefficients. For these reasons, in this thesis we have preferred the use of the direct learning approach for the DPD coefficients extraction.

Chapter 4

Dimensionality Reduction Techniques for PA Behavioral Modeling and DPD Linearization

The least squares solution for extracting the coefficients of PA behavioral model in (3.17) or for estimating the DPD coefficients increment in (3.38) can face the risk of poor performance, such as underfitting or overfitting. As the least squares issue in (3.17) is similar to the one in (3.38), from now on, we discuss about only the least squares in (3.17) and the same thing can be deduced for the one in (3.38). Figures 4.1a, 4.1b and 4.1c show examples for underfitting, overfitting and right fitting, respectively. Underfitting is the case when the model contains too few essential coefficients (i.e. the number of columns of the data matrix $\boldsymbol{X}$ is too small) to represent the data. Overfitting is the contrary effect, the model contains more coefficients than required (i.e. the number of columns of the data matrix $\boldsymbol{X}$ is too large) and is therefore unnecessarily complex. The large number of columns in the data matrix increases the possibility that some columns are linearly dependent among them. Thus, the Moore-Penrose inverse of the correlation matrix of $\boldsymbol{X}$ (i.e., $(\boldsymbol{X}^H\boldsymbol{X})^{-1}$) becomes close to singular. Consequently, the LS estimate tends to be highly sensitive to random errors (e.g. random noise, quantization noise of the measurement setup, etc.) in the observed response $\boldsymbol{y}$. Both underfitted and overfitted models result in misrepresenting the training data and introduce poor predictive performance. Thus, both of them have

to be avoided which should result in right fitting. In order to solve the underfitting effect, the order of the model can be increased. Whereas, to avoid the overfitting, regularization or dimensionality reduction techniques are employed.

Regularization is a process of introducing additional information in order to prevent overfitting. In general, the main idea of regularization techniques is to add a regularization term $\boldsymbol{R}(\boldsymbol{w})$ to the cost function.

$$J(\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \boldsymbol{R}(\boldsymbol{w}) \tag{4.1}$$

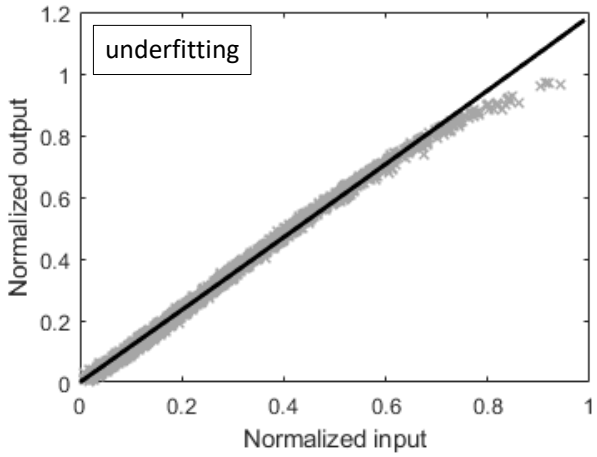
The regularization term $\boldsymbol{R}(\boldsymbol{w})$ is particularized according to the specific regularization technique selected. Regularization techniques such as, Ridge regression (or Tikhonov regularization) [71], the least absolute shrinkage and selection operator (LASSO) [72] or the Elastic Net [73] can be seen also as dimensionality reduction methods, since one of the consequences of regularization is the reduction of the number of coefficients. Analogously, considering the dimensionality reduction techniques that will be discussed in Section 4.1, by reducing the number of coefficients of the model a regularization effect is also introduced.

4.1 Dimensionality Reduction Techniques

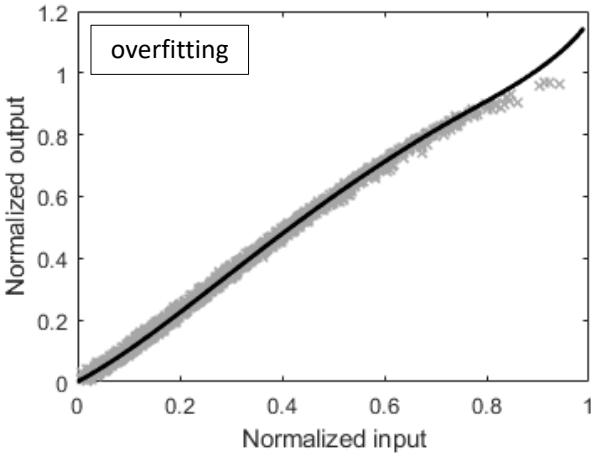
Dimensionality reduction techniques can help to avoid overfitting and ill-conditioning issues by properly reducing the number of basis functions, i.e., columns of the data matrix $\boldsymbol{X}$ for PA behavioral modeling in (3.17), or the data matrix $\boldsymbol{U}$ for DPD linearization in (3.38). These techniques remove redundant and irrelevant basis functions of the data matrix, retaining only most significant basis functions that contribute to model either the PA nonlinear behaviour or its inverse.

In general, dimensionality reduction techniques can be classified into two main groups:

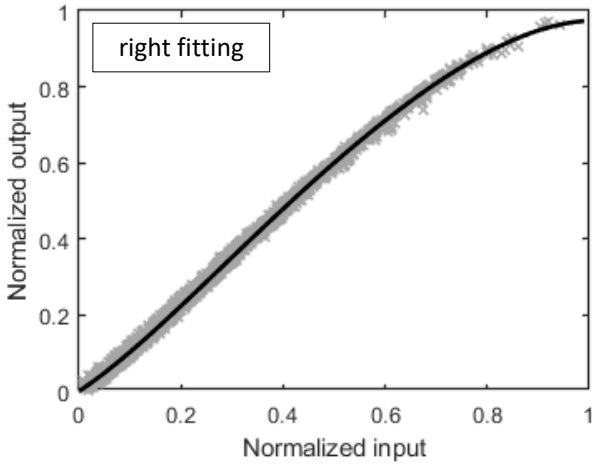
- **Feature selection techniques**: oriented at selecting the variables (i.e., basis



(a)



(b)



(c)

Figure 4.1: Underfitting, overfitting and right fitting in LS identification of the PA nonlinear behavior.

functions, regressors) from a set of original variables that are the most relevant for a particular model. Some examples of feature selection or regularization techniques that will be later addressed in Subsection 4.1.1 are: LASSO [74], Ridge regression [75], Elastic Net [73], matching pursuit [76] and orthogonal matching pursuit (OMP) [77].

- **Feature extraction techniques**: creating a reduced set of new variables that are linear or nonlinear combinations of the original variables. Some examples of this group are the principal component analysis (PCA) [78], partial least squares (PLS) [79] and canonical correlation analysis (CCA) [80]. Feature extraction techniques (PCA, PLS and CCA) will be discussed in Subsection 4.1.2.

Figure 4.2 shows the hierarchical structure of dimensionality reduction techniques.

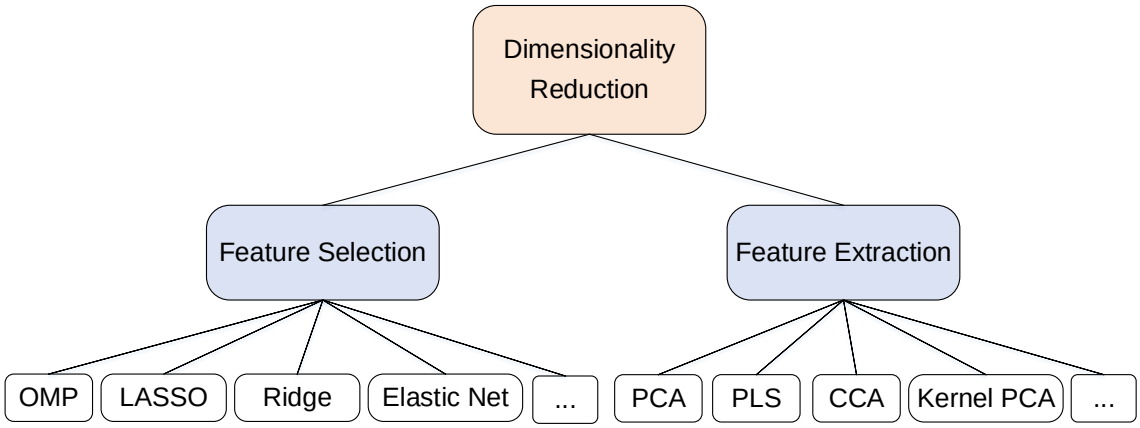


Figure 4.2: Hierarchical structure of dimensionality reduction techniques.

4.1.1 Feature Selection Techniques

As explained before, feature selection is the process of selecting the most relevant variables (i.e., basis functions or regressors) from a random set of original variables. The objective of feature selection techniques is to enforce the sparsity constraint on the vector of coefficients by minimizing the number of basis functions (i.e., ℓ_0-norm) subject to a constraint on the ℓ_2-norm squared of the identification error. For example,

particularizing for the identification of the PA behavioral model coefficients described in (3.17), the optimization problem can be described as

$$\begin{aligned} &\min_{w} \|\boldsymbol{w}\|_0 \\ &\text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \leq \varepsilon \end{aligned} \tag{4.2}$$

Unfortunately, this is a non-deterministic polynomial-time hard (NP-hard) combinatorial search problem. Therefore, in the field of DPD linearization, several sub-optimal approaches have been proposed targeting both robust identification and model order reduction.

In the following, some of the most popular feature selection techniques used in the field of DPD linearization will de presented. Therefore, on the one hand, further details will be given on the following regularization techniques: Ridge regression (or Tikhonov regularization), least absolute shrinkage and selection operator (LASSO) and Elastic Net. On the other hand, two popular greedy algorithms: the matching pursuit and its orthogonal variant, the OMP will be presented.

Ridge Regression or ℓ_2 Regularization

Ridge regression minimizes the sum of squares of the residual $\boldsymbol{e}$ in (3.15) subject to a constraint on the sum of squares of the coefficients, i.e.

$$\begin{aligned} &\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \\ &\text{subject to } \|\boldsymbol{w}\|_2^2 \leq t_2. \end{aligned} \tag{4.3}$$

or

$$\begin{aligned} &\min_{\boldsymbol{w}} \sum_{n=0}^{N-1} (y[n] - \boldsymbol{\varphi}_x^H[n]\boldsymbol{w}[n])^2 \\ &\text{subject to } \sum_{i=1}^{M} |w_i[n]|^2 \leq t_2. \end{aligned} \tag{4.4}$$

This constraint forces the coefficients to stay within a sphere of radius t_2 (as depicted in Figure 4.3). In Figure 4.3, the contours represent the values of coefficients estimated by the least squares solution. The coefficients under the Ridge regression are the ones on the contours that meet the constraint (i.e. the sphere).

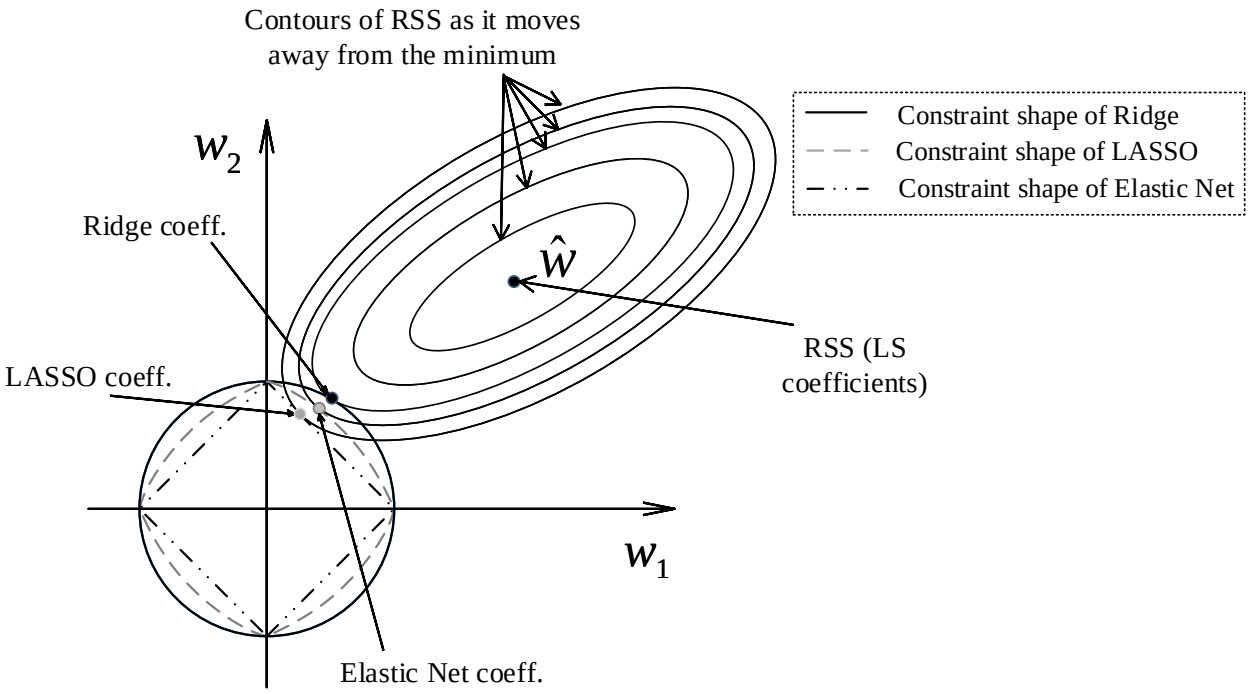


Figure 4.3: Ridge, LASSO and Elastic Net regularization.

The constrained cost function can also be written as a penalized residual sum of squares

$$\begin{aligned} J(\boldsymbol{w}) &= \sum_{n=0}^{N-1}(y[n]-\boldsymbol{\varphi}_x^H[n]\boldsymbol{w}[n])^2+\lambda_2\sum_{i=1}^{M}|w_i[n]|^2 \\ &= (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^H(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\lambda_2\|\boldsymbol{w}\|_2^2 \\ &= \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2+\lambda_2\|\boldsymbol{w}\|_2^2 \end{aligned} \tag{4.5}$$

where λ_2 ($\lambda_2 > 0$) is the shrinkage coefficient. Taking the derivative of the cost

function and setting it to zero we obtain the following solution,

$$\boldsymbol{w}_{Ridge} = (\boldsymbol{X}^H\boldsymbol{X} + \lambda_2\boldsymbol{I})^{-1}\boldsymbol{X}^H\boldsymbol{y}, \tag{4.6}$$

with $\boldsymbol{I}$ being the identity matrix. This approach enables to avoid the ill-conditioned issue in OLS since $(\boldsymbol{X}^H\boldsymbol{X} + \lambda_2\boldsymbol{I})$ is invertible even if $(\boldsymbol{X}^H\boldsymbol{X})$ is not [81]. As shown in (4.6), the Ridge coefficients $\boldsymbol{w}_{Ridge}$ depend on the shrinkage coefficient λ_2, i.e. as $\lambda_2 \to 0$, $\boldsymbol{w}_{Ridge}$ tends to the OLS solution; while as $\lambda_2 \to \infty$, $\boldsymbol{w}_{Ridge}$ tends to 0.

As an example, Figures 4.4 and 4.5 illustrate how the Ridge regression prevents the overfitting of PA modeling. The PA behavior is expressed using memory polynomial model. Figure 4.4-left shows the $NMSE$ of a PA nonlinear behavior identification with different model configurations of nonlinear order and memory. It can be seen that adding more memory or nonlinear terms does not allow the model achieve better $NMSE$, on the contrary, the $NMSE$ starts decreasing due to the ill-condition of the coefficient identification and consequently the estimated coefficients have high power values, as shown in Figure 4.5-left. By applying the Ridge, the power of the coefficients is restricted (as shown in Figure 4.5-right). Therefore, the $NMSE$ values are still stable though the model order keeps increasing (see Figure 4.4-right).

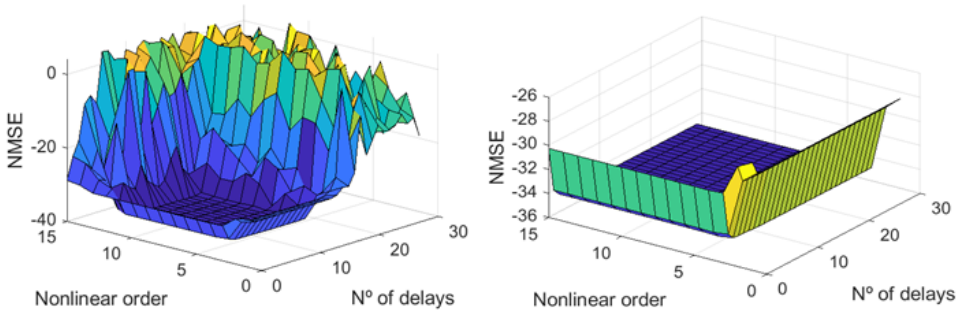


Figure 4.4: NMSE for different values of nonlinear order and memory taps when considering a memory polynomial model, without (left) and with (right) Tikhonov regularization.

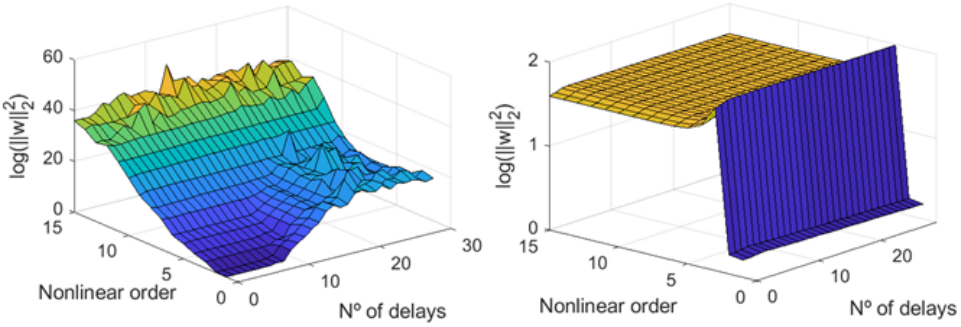


Figure 4.5: Squared norm of the vector of coefficients of the PA behavior identification for different values of nonlinear order and memory taps using a memory polynomial model, without (left) and with (right) Tikhonov regularization.

LASSO or ℓ_1 Regularization

Similarly to Ridge regression, LASSO [72] also gives a constraint to the cost function to limit the value of the model coefficients. While the constraint of the Ridge regression is the sum of square of the coefficients, the LASSO constraint is the sum of the absolute value of the coefficients.

$$\begin{aligned} &\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \\ &\text{subject to } \|\boldsymbol{w}\|_1 \leq t_1. \end{aligned} \tag{4.7}$$

or

$$\begin{aligned} &\min_{\boldsymbol{w}} \sum_{n=0}^{N-1} (y[n] - \boldsymbol{w}^T[n]\boldsymbol{\varphi}_x[n])^2 \\ &\text{subject to } \sum_{i=1}^{M} |w_i[n]| \leq t_1. \end{aligned} \tag{4.8}$$

This constraint forces the coefficients to stay within the diamond shape (see Figure 4.3). The constrained cost function can also be written as a penalized residual sum

of squares

$$\begin{aligned} J(\boldsymbol{w}) &= \sum_{n=0}^{N-1}(y[n]-\boldsymbol{\varphi}_x^H[n]\boldsymbol{w}[n])^2+\lambda_1\sum_{i=1}^{M}|w_i[n]| \\ &= (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^H(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\lambda_1\|\boldsymbol{w}\|_1 \\ &= \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2+\lambda_1\|\boldsymbol{w}\|_1 \end{aligned} \tag{4.9}$$

where λ_1 ($\lambda_1 > 0$) is the shrinkage coefficient. The regression coefficients are estimated as

$$\boldsymbol{w}_{LASSO} = (\boldsymbol{X}^H\boldsymbol{X})^{-1}(\boldsymbol{X}^H\boldsymbol{y}-\frac{\lambda_1}{2}\boldsymbol{b}), \tag{4.10}$$

where the elements b_i of $\boldsymbol{b}$ are either $+1$ or -1, depending on the sign of the corresponding regression coefficient $w_i[n]$. Despite the fact that the original implementation involves quadratic programming techniques from convex optimization, Efron et al. in [82] proposed the least angle regression (LARS) algorithm that can be used for computing the LASSO path efficiently.

Elastic Net

When considering a model with the number of components (or basis functions) M is bigger than the number of observations N (i.e., $M > N$), LASSO tends to select all components. Moreover, in case there is a group of highly correlated components, LASSO selects only one component from the group and ignores the others. In order to overcome these limitations of LASSO, Elastic Net [73] was proposed to add one more constraint into the cost function of LASSO. That is the constraint of Ridge. The Elastic Net constraints are depicted in Figure 4.3.

$$\begin{aligned} &\min_{\boldsymbol{w}} \sum_{n=0}^{N-1}(y[n]-\boldsymbol{w}^T[n]\boldsymbol{\varphi}_x[n])^2 \\ &\text{subject to } \sum_{i=1}^{M}|w_i[n]|^2 \le t_2 \text{ and } \sum_{i=1}^{M}|w_i[n]| \le t_1. \end{aligned} \tag{4.11}$$

The constrained cost function can also be written as a penalized residual sum of squares

$$\begin{aligned} J(\boldsymbol{w}) &= \sum_{n=0}^{N-1}(y[n]-\boldsymbol{\varphi}_x^H[n]\boldsymbol{w}[n])^2+\lambda_2\sum_{i=1}^{M}|w_i[n]|^2+\lambda_1\sum_{i=1}^{M}|w_i[n]| \\ &= (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^H(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\lambda_2\|\boldsymbol{w}\|_2^2+\lambda_1\|\boldsymbol{w}\|_1 \\ &= \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2+\lambda_2\|\boldsymbol{w}\|_2^2+\lambda_1\|\boldsymbol{w}\|_1 \end{aligned} \tag{4.12}$$

where $\lambda_2 > 0$ and $\lambda_1 > 0$ are the shrinkage coefficients. For the Elastic Net the regression coefficients are estimated as

$$\boldsymbol{w}_{E-net} = (\boldsymbol{X}^H\boldsymbol{X}+\lambda_2\boldsymbol{I})^{-1}(\boldsymbol{X}^H\boldsymbol{y}-\frac{\lambda_1}{2}\boldsymbol{b}), \tag{4.13}$$

The minimization of the elastic net cost function in (4.12) is similar to minimizing the LASSO cost function and the entire elastic net regularization paths can be estimated almost as efficiently as the LASSO paths with the LARS-EN algorithm proposed in [73].

Matching Pursuit

The greedy iterative algorithm matching pursuit was proposed by Mallat and Zhang in 1992 [83]. In case the basis matrix $\boldsymbol{X}$ of the PA behavioral model is over-dimensioned or redundant, matching pursuit can be applied to find the most essential columns of $\boldsymbol{X}$.

The matching pursuit algorithm selects, at each iteration step, the most relevant column, defined as the one that most strongly correlates with the current residual error $\boldsymbol{r}$ (initialized to $\boldsymbol{r}^{(0)} = \boldsymbol{y}$). The covariance is identified by the inner products between the residual and each column of the matrix of basis functions $\boldsymbol{X}$. Then, the residual error is updated to seek the next most important column. With matching pursuit, a column may be considered multiple times.

The matching pursuit algorithm is summarized in Algorithm 1. The number of most important columns K we wish to select is given as an input of the algorithm. Algorithm 1 returns the support set $\boldsymbol{S}$ that contains the indices of all the columns in $\boldsymbol{X}$. The columns are sorted based on the descending trend of the relevance of each column. Therefore, the first K columns are the most important ones.

Algorithm 1 Matching Pursuit

procedure MPA($\boldsymbol{X}, \boldsymbol{y}, K$)
    initialization:
    $\boldsymbol{r}^{(0)} = \boldsymbol{y}; \quad \boldsymbol{S}(0) = \{\}; \quad n = 0;$
    for $n = 1$ to K **do**
        $i^{(n)} \leftarrow \text{argmax}_i |\boldsymbol{X}_{\{i\}}^H \boldsymbol{r}^{(n-1)}|;$
        $\boldsymbol{S}(n) \leftarrow \boldsymbol{S}(n-1) \bigcup \{i^{(n)}\};$
        $\boldsymbol{a}^n \leftarrow \boldsymbol{X}_{\{i\}}^H \boldsymbol{r}^{(n-1)};$
        $\boldsymbol{r}^{(n)} \leftarrow \boldsymbol{r}^{(n-1)} - \boldsymbol{a}^n \boldsymbol{X}_{\{i\}};$
    end for
    Return $\boldsymbol{S}$
end procedure

Orthogonal Matching Pursuit

Orthogonal matching pursuit described by C. Pati et al. in 1993 in [84] is a modification of the matching pursuit algorithm. OMP is similar to the matching pursuit except for the fact that, at each step of the algorithm, the residual error $\boldsymbol{r}$ is updated taking into account the contribution of the already selected columns (their associated coefficients are extracted via LS identification). Consequently, the residual error is orthogonal to the already chosen columns. With this scheme, OMP gives better sorting result than matching pursuit at the price of introducing higher complexity since a LS identification is calculated at every iteration of the algorithm.

Algorithm 2 summarizes the OMP algorithm. The number of basis functions under consideration m_{max} is the maximum number of columns of $\boldsymbol{X}$. The support set containing the indices of the selected columns at m_{th} iteration is noted as $\boldsymbol{S}^{(m)}$. At each m_{th} iteration of the OMP algorithm, the basis function $\boldsymbol{X}_{\{i\}}$ that best contributes

to minimize the residual error $\boldsymbol{e}^{(m-1)}$ is selected based on the covariance between the basis function and the residual error. The selected basis function is then added to the support set $\boldsymbol{S}^{(m)}$. At the end, the algorithm returns the vector $\boldsymbol{S}^{(m_{max})}$ that includes the indices of all original basis functions sorted according to their relevance. In order to determine the optimum number of essential basis functions, the OMP is commonly combined with some information criterion such as the Bayesian information criterion (BIC) [77] or the Akaike information criterion (AIC) [85].

Algorithm 2 Orthogonal Matching Pursuit Algorithm

procedure OMP($\boldsymbol{y}, \boldsymbol{X}, m_{max}$)
  initialization:
  $\boldsymbol{e}^{(0)} = \boldsymbol{y} - \widehat{\boldsymbol{y}}^{(0)}$; with $\widehat{\boldsymbol{y}}^{(0)} = 0$
  $\boldsymbol{S}^{(0)} = \{\}$
  for $m = 1$ **to** m_{max} **do**
    $i^{(m)} = \arg\min_{i} \min_{w_i} ||\boldsymbol{e}^{(m-1)} - \boldsymbol{X}_{\{i\}} w_i||_2^2 \approx \arg\max_{i} \left| \boldsymbol{X}_{\{i\}}^H \boldsymbol{e}^{(m-1)} \right|$
    $\boldsymbol{S}^{(m)} \leftarrow \boldsymbol{S}^{(m-1)} \bigcup i^{(m)}$
    $\boldsymbol{w}^{(m)} = \left(\boldsymbol{X}_{S^{(m)}}^H \boldsymbol{X}_{S^{(m)}} \right)^{-1} \boldsymbol{X}_{S^{(m)}}^H \boldsymbol{y}$
    $\widehat{\boldsymbol{y}}^{(m)} = \boldsymbol{X}_{S^{(m)}} \boldsymbol{w}^{(m)}$
    $\boldsymbol{e}^{(m)} = \boldsymbol{y} - \widehat{\boldsymbol{y}}^{(m)}$
  end for
  return $\boldsymbol{S}^{(m_{max})}$
end procedure

4.1.2 Feature Extraction Techniques

Feature extraction techniques first generate a set of new components (i.e., basis functions) which are linear or nonlinear combinations of the original data. Then, the most significant components of the new basis are retained while the irrelevant ones are removed. This Subsection focuses on PCA, PLS and CCA techniques.

Principal Component Analysis

Principal component analysis (PCA) is a statistical learning technique introduced by K. Pearson in 1901 [86], suitable for converting an original basis of eventually correlated features or components into a new uncorrelated orthogonal basis set called principal components. The principal components are linear combinations of the original variables oriented to capture the maximum variance in the data.

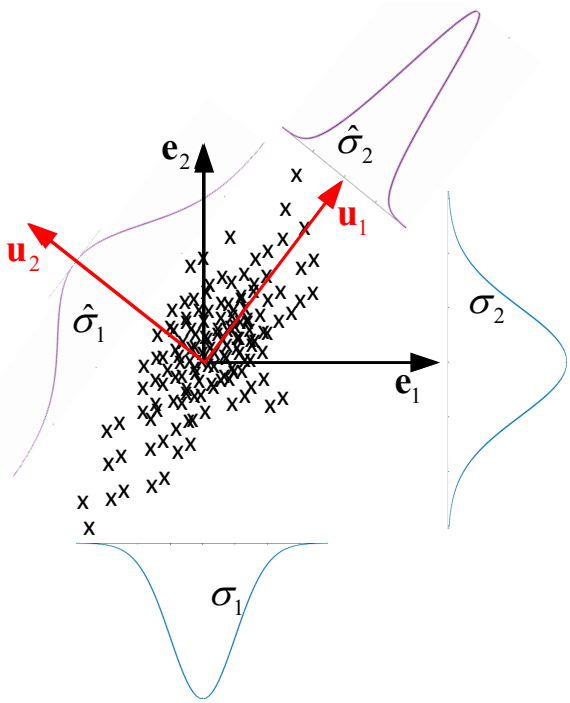


Figure 4.6: PCA transformation considering 2-dimensional data.

Figure 4.6 presents an example of PCA transformation in a 2-dimensional data. The original coordinate axes are $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$. The new coordinate axes are $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ corresponding to the two eigenvectors of the original data. The first eigenvector (corresponding to axis $\boldsymbol{u_1}$) has much larger variance than the second eigenvector (corresponding to axis $\boldsymbol{u_2}$) (i.e. $\widehat{\sigma}_1 > \widehat{\sigma}_2$). Whereas, in the original coordinate axes, the difference of the variances of the data on the two axes are not significantly different. In this example, $\boldsymbol{u_1}$ is the principal component of the considered data. Therefore,

if we discard the unimportant component $\boldsymbol{u_2}$ and project the data on the dimension $\boldsymbol{u_1}$, the information loss will not be very significant. Whereas, in the original coordinate axes, if we simply remove one dimension and retain the other, we will encounter relevant information loss.

Partial Least Squares

PLS was introduced by H. O. Wold [87]. Similar to PCA, PLS is a statistical technique used to construct a new basis of components that are linear combinations of the original basis functions. However, while PCA obtains new components that maximize their own variance, PLS finds linear combinations of the original variables that maximize the covariance between the new components and the reference data. This difference enables PLS to outperform PCA in applications such as dimensionality reduction for PA behavioral modeling and DPD linearization as will be presented later in Subsection 6.2.3.

Canonical Correlation Analysis

The CCA method, introduced by H. Hotelling in 1936 [80], finds linear combinations of variables of a given data that maximally correlate to the reference data. Same as PCA and PLS, CCA is also widely used to reduce the dimension of a given data. The main differences among them are discussed in the following.

- PCA finds new orthogonal components with maximal *variance* among themselves. PCA takes into account only the input data. Its performance is independent from the reference data (in our considered DPD problem, the reference data is the output signal of PA $\boldsymbol{y}$ or the residual linearization error $\boldsymbol{e}$).

- PLS finds linear combinations of the original basis functions that maximize the *covariance* between the new components and the reference data. This enables

PLS to outperform PCA in applications such as dimensionality reduction for PA behavioral modeling and DPD linearization

- CCA finds new components with maximal *correlation* between the new components and the reference data. Same as PLS, CCA also depends on both the original and the reference data. Since the correlation relationship among the components does not depend on the length of the components (unlike the covariance relationship), CCA shows better performance than PLS.

A comparison of the three feature extraction techniques presented in this Subchapter is summarized in Table 4.1. As can be seen from Table 4.1, the different statistical measures used in three given techniques result different reduction performance. CCA is the best among three dimensionality reduction techniques. The second best technique is PLS and the last one is PCA. Whereas PLS has lowest computational cost. PCA has highest complexity and CCA is in the middle. Therefore, in Chapter 7, we propose to use a combination of PCA (calculated off-line only once) and a modified PLS to achieve the performance same as the CCA's but with the computational cost of PLS.

Table 4.1: Comparison among PCA, PLS and CCA. N: number of samples and M: dimension of the given data.

Method	Statistical measure	Reference signal	Complexity	Reduction performance
PCA	Variance	No	$O(NM^2 + M^3)$ [88]	3^{rd}
PLS	Covariance	Yes	$O(NM)$ [89, 90]	2^{nd}
CCA	Correlation	Yes	$O(NM^2)$ [91]	1^{st}

4.2 Dimensionality Reduction of the Digital Predistortion Linearization System

In the field of DPD linearization, dimensionality reduction techniques are used with a double objective. On the one hand, to ensure a proper well-conditioned coefficient identification and, on the other hand, to reduce the number of coefficients to be estimated and thus relaxing the computational complexity and memory requirements of a hardware implementation.

As presented in Section 3.4, an adaptive DPD linearization system is composed of two subsystems: the forward path and the feedback path. The DPD coefficients reduction can be implemented for both the DPD forward path and feedback path subsystems. Generally, feature selection techniques are applied in the forward path to select the most significant coefficients to be used in the DPD function, while feature extraction techniques are employed in the adaptation subsystem to reduce the number of DPD coefficients to be estimated.

4.2.1 Order reduction of the DPD Function in the Forward Path

As explained before, in the field of DPD linearization, several approaches have been proposed targeting both robust identification and model order reduction. For example, LASSO was used by Wisell et al. in [92] consisting in a ℓ_1-norm regularization; the Ridge regression was used for example by Guan et al. in [93] consisting in a ℓ_2-norm regularization; the sparse Bayesian learning (SBL) algorithm was used by Peng et al. in [94]; or the orthogonal matching pursuit was used in [77] by Reina et al. to select the most relevant basis functions of the DPD function.

Note that in order to minimize the number of coefficients being required by the DPD function in the forward path, we assume that the optimal subset of selected basis functions of the DPD function will be the same as that used for PA behavioral

modeling. Therefore, we can perform the feature selection on the matrix of basis functions $\boldsymbol{X}$ in (3.12) since the selected basis functions used for PA behavioral modeling will be later used to build matrix $\boldsymbol{U}$ in (3.35) for DPD linearization purposes.

The main contribution on this thesis is not focused in the feature selection techniques for dimensionality reduction of the DPD function in the forward path. For that reason, in our research we simply employed the OMP algorithm (presented in Algorithm 2) to reduce the order of the PA behavioral model and DPD function since OMP showed good reduction vs. modeling accuracy performance. Therefore, the OMP search will be carried out off-line once and then the reduced set of basis functions will be used for DPD linearization with two purposes: reducing the number of coefficients of the forward path DPD behavioral model and improving the conditioning and robustness of the adaptation subsystem.

4.2.2 Simplification of the Identification/Adaptation Algorithm in the DPD Feedback Path

The main scope of this thesis is the proposal of several feature extraction dimensionality reduction techniques for PA behavioral modeling and DPD coefficient estimation/adaptation. The DPD dimensionality reduction is carried out by calculating a new reduced set of orthogonal components that are linear combinations of the original basis functions. Note that applying feature extraction techniques, we reduce only the number of estimated coefficients in DPD feedback path subsystem, but the number of coefficients in the DPD function in the forward path remain unaltered. For that reason, in the next three Chapters of this thesis, the proposed dimensionality reduction strategies for DPD coefficient estimation/adaptation will be properly combined with the OMP feature selection method in the forward path.

In either commercial products or publications addressing DPD implementation, QR factorization is the most common solution to solve the LS regression problem. For that reason, we will take the QR-LS identification/adaptation method as baseline

for comparison with the proposed feature extraction strategies in next Chapters.

This Subsection provides a general description of the benefits of using feature extraction techniques for the DPD identification/adaptation, particularizing for PCA and PLS, where the new orthogonal transformed matrix significantly simplifies the LS coefficients extraction.

DPD coefficients extraction using PCA or PLS

The objective of using PCA in DPD linearization is to transform the $N \times M$ matrix of basis functions $\boldsymbol{U}$ in (3.38) into a new set of orthogonal components. The $N \times L$ (with $L \leq M$) transformed matrix $\widehat{\boldsymbol{U}}_{pca}$ contains only the most relevant components (or new basis functions) of $\boldsymbol{U}$. Thus, the less relevant or redundant components can be removed, which reduces the complexity of the DPD coefficients adaptation and avoids ill-conditioning.

The transformation of the basis matrix $\boldsymbol{U}$ into $\widehat{\boldsymbol{U}}_{pca}$ is carried out via the $M \times L$ *transformation matrix* $\boldsymbol{P}_{pca}$ as follows

$$\widehat{\boldsymbol{U}}_{pca} = \boldsymbol{U}\boldsymbol{P}_{pca} \tag{4.14}$$

The transformation matrix $\boldsymbol{P}_{pca}$ contains the most important eigenvectors of the covariance matrix of $\boldsymbol{U}$ ($cov(\boldsymbol{U})$). The covariance matrix of $\boldsymbol{U}$ can be approximated by the correlation matrix of $\boldsymbol{U}$ ($\boldsymbol{U}^H\boldsymbol{U}$) because the expectation of $\boldsymbol{U}$ ($E\{\boldsymbol{U}\}$) is 0:

$$cov(\boldsymbol{U}) = \frac{1}{N-1}\big((\boldsymbol{U} - E\{\boldsymbol{U}\})^H(\boldsymbol{U} - E\{\boldsymbol{U}\})\big) \approx \boldsymbol{U}^H\boldsymbol{U}. \tag{4.15}$$

Whereas, the principal components of the basis functions (i.e., columns of $\boldsymbol{U}$) are the eigenvectors of $\boldsymbol{U}\boldsymbol{U}^H$.

The matrix $\widehat{\boldsymbol{U}}_{pca}$ is then used in (3.38) instead of $\boldsymbol{U}$ to compute the $L \times 1$ vector of transformed DPD coefficients increment $\Delta\hat{\boldsymbol{w}}$, as described in (4.17). The new transformed matrix $\widehat{\boldsymbol{U}}_{pca}$ contains the most relevant components sorted in descending

order of importance, i.e., the most relevant one is in the first column of $\widehat{\boldsymbol{U}}_{pca}$. The coefficients update is performed calculating the vector $\Delta\hat{\boldsymbol{w}}$ which contains only L elements instead of the M elements (with $L \leq M$) of the vector $\Delta\boldsymbol{w}$. Then, after finding $\Delta\hat{\boldsymbol{w}}$, it is possible to go back to the original vector $\boldsymbol{w}$ that will be used in DPD function in the forward path.

Similarly to PCA, PLS is used to create a new basis matrix $\widehat{\boldsymbol{U}}_{pls}$ of $N \times L$ orthonormal components from the $N \times M$ original data matrix $\boldsymbol{U}$ through a $M \times L$ transformation matrix $\boldsymbol{P}_{pls}$ as follows,

$$\widehat{\boldsymbol{U}}_{pls} = \boldsymbol{U}\boldsymbol{P}_{pls}. \tag{4.16}$$

The transformation matrix $\boldsymbol{P}_{pls}$ is obtained by using an iterative algorithm named SIMPLS proposed by De Jong, S. in 1993 in [95]. The new orthonormal components of the transformed matrix $\widehat{\boldsymbol{U}}_{pls}$ are sorted according to their contribution to maximize the covariance between the new components and the reference signal which, in the case of DPD linearization, the reference signal is the error signal $\boldsymbol{e}$ defined in (3.39).

Taking into account the new transformed matrix $\widehat{\boldsymbol{U}}$ (where $\widehat{\boldsymbol{U}}$ refers to either $\widehat{\boldsymbol{U}}_{pca}$ or $\widehat{\boldsymbol{U}}_{pls}$) with orthogonal or orthonormal components, the extraction of the vector of transformed DPD coefficients increment using the direct learning approach in (3.39) turns to

$$\boldsymbol{\Delta}\widehat{\boldsymbol{w}} = \left(\widehat{\boldsymbol{U}}^H\widehat{\boldsymbol{U}}\right)^{-1}\widehat{\boldsymbol{U}}^H\boldsymbol{e}. \tag{4.17}$$

Taking into account that with PCA the components of the transformed matrix $\widehat{\boldsymbol{U}}_{\boldsymbol{pca}}$ are orthogonal among them, the correlation matrix inversion of the LS solution remains as follows,

$$\left(\widehat{\boldsymbol{U}}_{\boldsymbol{pca}}{}^H\widehat{\boldsymbol{U}}_{\boldsymbol{pca}}\right)^{-1} = diag\left(\lambda_1^{-1}, \cdots, \lambda_j^{-1} \cdots, \lambda_L^{-1}\right) \tag{4.18}$$

with λ_j being the eigenvalues of $\boldsymbol{U}^H\boldsymbol{U}$ and $\boldsymbol{U}\boldsymbol{U}^H$. While, in the case of using PLS,

the transformed matrix $\widehat{\boldsymbol{U}}_{\boldsymbol{pls}}$ contains orthonormal components and thus

$$\left(\widehat{\boldsymbol{U}}_{\boldsymbol{pls}}{}^{H}\widehat{\boldsymbol{U}}_{\boldsymbol{pls}}\right)^{-1} = \boldsymbol{I}. \tag{4.19}$$

Consequently, the computation of the DPD coefficients increment $\Delta\hat{\boldsymbol{w}}$ in (4.17) is significantly simplified. Finally, once the vector $\boldsymbol{\Delta}\widehat{\boldsymbol{w}}$ is obtained, it can be converted back to $\boldsymbol{\Delta w}$ by using the transformation matrix $\boldsymbol{P}$ (denoting either $\boldsymbol{P}_{pca}$ or $\boldsymbol{P}_{pls}$).

$$\boldsymbol{\Delta w} = \boldsymbol{P}\boldsymbol{\Delta}\widehat{\boldsymbol{w}} \tag{4.20}$$

where $\boldsymbol{\Delta w}$ will be used in (3.37) to update the new DPD coefficients.

Chapter 5

DPD Coefficients Estimation/Adaptation using Block Deflated Adaptive Principal Component Analysis

5.1 Introduction

The data matrix $\boldsymbol{U}$ may have large variation since the characteristics of PA can change over time. The conventional PCA (presented in Subsection 4.2.2) cannot deduce efficiently the principal components of such data [96]. Hence, in this case, an adaptive PCA scheme is required to compute the principal components online. APCA adaptively generates the transformation matrix $\boldsymbol{P}_{pca}$ without requesting the correlation matrix $\boldsymbol{U}^H\boldsymbol{U}$ in advance. APCA is suitable for real time application and can be easily implemented in embedded processors because it has low requirement for storage memory and reduced computational cost.

This Chapter presents a method, based on the APCA technique, that iteratively creates and updates an orthogonal data matrix used to estimate the coefficients of PA behavioral models or DPD linearizers. The method is named block deflated APCA (BD-APCA). BD-APCA is designed by properly modifying the well-known complex domain generalized Hebbian algorithm (CGHA). The generalized Hebbian algorithm

(GHA) and CGHA are presented in Section 5.2. In Section 5.3 the BD-APCA technique is described in detail. This adaptation method enhances the robustness of the coefficient estimation, simplifies the adaptation by reducing the number of estimated coefficients and due to the orthogonality of the new basis, these coefficients can be estimated independently, thus allowing for scalability. Experimental results in Section 5.4 will show that the proposed BD-APCA method is a worthy solution for online, adaptive and robust coefficient estimation for PA modeling and DPD linearization with a reduced number of required coefficients.

5.2 Generalized Hebbian Algorithm

The GHA is a linear feedforward neural network model for unsupervised learning introduced by Sanger in [97] that is used to iteratively find the transformation matrix of PCA. In order to apply GHA taking into account complex numbers, Zhang et al. proposed the complex domain GHA (CGHA) in [98].

5.2.1 Generalized Hebbian Algorithm for Real Signals

The key point of PCA is to find principal eigenvectors that are associated with the largest eigenvalues. Generally, the correlation matrix $\boldsymbol{U}^H\boldsymbol{U}$ is required as the input to compute the eigenvectors in the transformation matrix of the PCA. However, the iterative scheme of GHA allows to deduce the eigenvectors directly from the vectors of the data matrix $\boldsymbol{U}$, without the need of the correlation matrix. The input to the GHA is the vector $\boldsymbol{\varphi_u}$, which is defined as the transpose of a row in the data matrix $\boldsymbol{U}$ (containing the basis functions for DPD linearization),

$$\boldsymbol{\varphi_u} = \left(\phi_1^u, \cdots, \phi_i^u, \cdots, \phi_M^u\right)^T \tag{5.1}$$

where ϕ_i^u is the original i^{th} basis function, with $i = 1, \cdots, M$. The output of GHA is the transformation matrix $\boldsymbol{P}_{pca}$ that includes L principal eigenvectors:

$$\begin{aligned} \boldsymbol{p}_1 = &\quad (p_{11},\ p_{12}, \cdots,\ p_{1M})^T \\ \boldsymbol{p}_2 = &\quad (p_{21},\ p_{22}, \cdots,\ p_{2M})^T \\ &\quad ... \\ \boldsymbol{p}_L = &\quad (p_{L1},\ p_{L2}, \cdots,\ p_{LM})^T \end{aligned} \tag{5.2}$$

First, the initial values of $\boldsymbol{p}_l$ $(l = 1, 2, ..., L)$ are randomly set. Then, the eigenvectors are iteratively updated as follows

$$\boldsymbol{p}_l^{(j+1)} = \boldsymbol{p}_l^{(j)} + \mu^{(j)} y_l^{(j)} \left[\boldsymbol{\varphi}_u^{(j)} - y_l^{(j)} \boldsymbol{p}_l^{(j)} - \sum_{i<l} y_i^{(j)} \boldsymbol{p}_i^{(j)} \right] \tag{5.3}$$

where $y_l^{(j)} = \left(\boldsymbol{p}_l^{(j)} \right)^T \boldsymbol{\varphi_u}^{(j)}$, j is the iteration index and $\mu^{(j)}$ the learning rate factor. After a number of iterations (depending on the learning rate factor), $\boldsymbol{p}_l$ converges to the l^{th} eigenvector of $\boldsymbol{U}^H \boldsymbol{U}$.

5.2.2 Generalized Hebbian Algorithm for Complex Signals

The complex GHA (CGHA) is similar to GHA but the operations for real data are replaced with the ones for complex data. Therefore, the update for each vector $\boldsymbol{p}_l$ of the transformation matrix $\boldsymbol{P}_{pca}$ is

$$\boldsymbol{p}_l^{(j+1)} = \boldsymbol{p}_l^{(j)} + \mu^{(j)} (y_l^{(j)})^* \left[\boldsymbol{\varphi}_u^{(j)} - y_l^{(j)} \boldsymbol{p}_l^{(j)} - \sum_{i<l} y_i^{(j)} \boldsymbol{p}_i^{(j)} \right] \tag{5.4}$$

in which $(y_l^{(j)})^*$ is the complex conjugate of $y_l^{(j)} = \left(\boldsymbol{p}_l^{(j)} \right)^H \boldsymbol{\varphi_u}^{(j)}$. The advantages of using CGHA to compute the matrix $\boldsymbol{P}_{pca}$ are:

- CGHA avoids the expensive calculation of the correlation matrix $\boldsymbol{U}^H \boldsymbol{U}$. It does

not require the full data matrix $\boldsymbol{U}$ in advance.

- CGHA is expandable. That is, the updating of the $(L+1)^{th}$ eigenvector does not affect the preceding L vectors.

Algorithm 3 BD-APCA Algorithm

procedure BD-APCA($\boldsymbol{U}, L$)
  initialization:
  for $j = 1$ to L **do**
    $\boldsymbol{r}_j[0] = rand()$;
  end for
  for $n = 0$ to $N-1$ **do**
    $\boldsymbol{b}_1[n] = \boldsymbol{\varphi}_u[n]$;
  end for
  for $j = 1$ to L **do**
    $\boldsymbol{B}_j = (\boldsymbol{b}_j[0], \boldsymbol{b}_j[1], \cdots, \boldsymbol{b}_j[N-1])^T$;
    $k = 0$;
    $\boldsymbol{r}_{j,k+1}[0] = \boldsymbol{r}_j[0]$;
    repeat
      $k = k + 1$;
      for $n = 0$ to $N-1$ **do**
        $d_{j,k}[n] = \boldsymbol{r}_{j,k}^H[n]\boldsymbol{b}_j[n]$;
        $\eta_{j,k} = \sigma \cdot trace(\boldsymbol{B}_j^H \boldsymbol{B}_j)/k$;
        $\boldsymbol{r}_{j,k}[n+1] = \boldsymbol{r}_{j,k}[n] + \eta_{j,k}(d_{j,k}[n])^* \big(\boldsymbol{b}_j[n] - d_{j,k}[n]\boldsymbol{r}_{j,k}[n]\big)$;
      end for
    until $\boldsymbol{r}_{j,k}$ *steady*;
    $\boldsymbol{r}_j = \boldsymbol{r}_{j,k}[N]$;
    for $n = 0$ to $N-1$ **do**
      $\boldsymbol{b}_{j+1}[n] = \boldsymbol{b}_j[n] - \big(\boldsymbol{r}_j^H \boldsymbol{b}_j[n]\big)\boldsymbol{r}_j$;
    end for
  end for
  Return $\boldsymbol{R} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_L)$
end procedure

5.3 Block Deflated Adaptive Principal Component Analysis

Based on CGHA, the block deflated adaptive principal component analysis (BD-APCA) algorithm iteratively finds the $M \times L$ transformation matrix $\boldsymbol{R} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_L)$, with $L \leq M$. After a number of iterations, the matrix $\boldsymbol{R}$ will finally converge to the matrix $\boldsymbol{P}_{pca}$ in (4.14) without the requirement of calculating the correlation matrix $\boldsymbol{U}^H\boldsymbol{U}$. In comparison to CGHA, the BD-APCA algorithm presents the following differences:

- The updated data in the BD-APCA is considered as a block of data vectors, instead of only one vector as in CGHA.
- In BD-APCA, the columns $\boldsymbol{r}_j$ $(j = 1, 2, \cdots, L)$ of the transformation matrix $\boldsymbol{R}$ are found iteratively one by one. Whereas, in CGHA, all the columns of the transformation matrix are calculated at the same time. In BD-APCA, by processing the columns separately in a sequential fashion, the number of iterations of the estimation and the learning-rate coefficient η of each column can be properly controlled. Besides, the next column is estimated by using the values of the previously extracted components. These advantages enable BD-APCA to enhance both the convergence time and the final performance.
- The BD-APCA uses variable learning-rate coefficient η for each column $\boldsymbol{r}_j$ $(j = 1, 2, \cdots, L)$. As will be shown later, the η coefficients are computed from the deflated data.

Algorithm 3 summarizes the BD-APCA method. The inputs of the algorithm are the data matrix $\boldsymbol{U} = \Big(\boldsymbol{\varphi}_u[0], \boldsymbol{\varphi}_u[1], \cdots, \boldsymbol{\varphi}_u[n], \cdots, \boldsymbol{\varphi}_u[N-1]\Big)^T$ (see (3.36)) and the dimensionality L of the desired transformation matrix $\boldsymbol{R}$. The algorithm results the transformation matrix $\boldsymbol{R}$ which is used as the transformation matrix $\boldsymbol{P}_{pca}$ for model order reduction in both PA behavioral modeling and DPD linearization applications.

The operation * in line 18 of the algorithm denotes the complex conjugate. The learning rate η used to update each column of $\boldsymbol{R}$ is obtained as a proportional factor of the power of the input data by using the $trace(\boldsymbol{B}^H\boldsymbol{B})$. The $trace(\cdot)$ is the sum of the vector's modulus. σ in line 17 is a constant factor.

5.4 Experimental Test Bench and Results

5.4.1 Experimental Test Bench

The BD-APCA technique is validated for PA behavioral modeling and DPD linearization using the Matlab–controlled digital linearization test bench. The test bench is depicted in Figure 5.1.

In the test bench, four 20-MHz bandwidth and 64-QAM modulated FC-FBMC signals, each with different sub-carrier group deactivation configurations, have been carrier aggregated. The overall test signals feature 80 MHz bandwidth and around 13 dB PAPR. Considering a DPD expansion factor by three, the DPD baseband waveform length is of 737280 samples.

For signal generation and data capture, we used commercial boards from Texas Instruments (TI). Specifically, TI TSW1400EVM and TSW30H84EVM at Tx side and TI ADC32RF45EVM and TSW14J56EVM at Rx side. In order to account for the out-of-band distortion, a 368.64 MSa/s DPD signal with 240 MHz bandwidth was up converted to the 875 MHz RF frequency to feed a class-J PA based on the Cree CGH35030F GaN HEMT. The PA output signal (with +28 dBm mean output power) was attenuated, RF sampled at 2457.6 MSa/s, and further down-sampled to the DPD signal sample rate for time-alignment and DPD processing. The class-J PA under test was designed by the research group of Dr. Jose Angel García at the University of Cantabria, while the test bench was assembled by David López-Bueno at the Centre Tecnològic de Telecomunicacions de Catalunya.

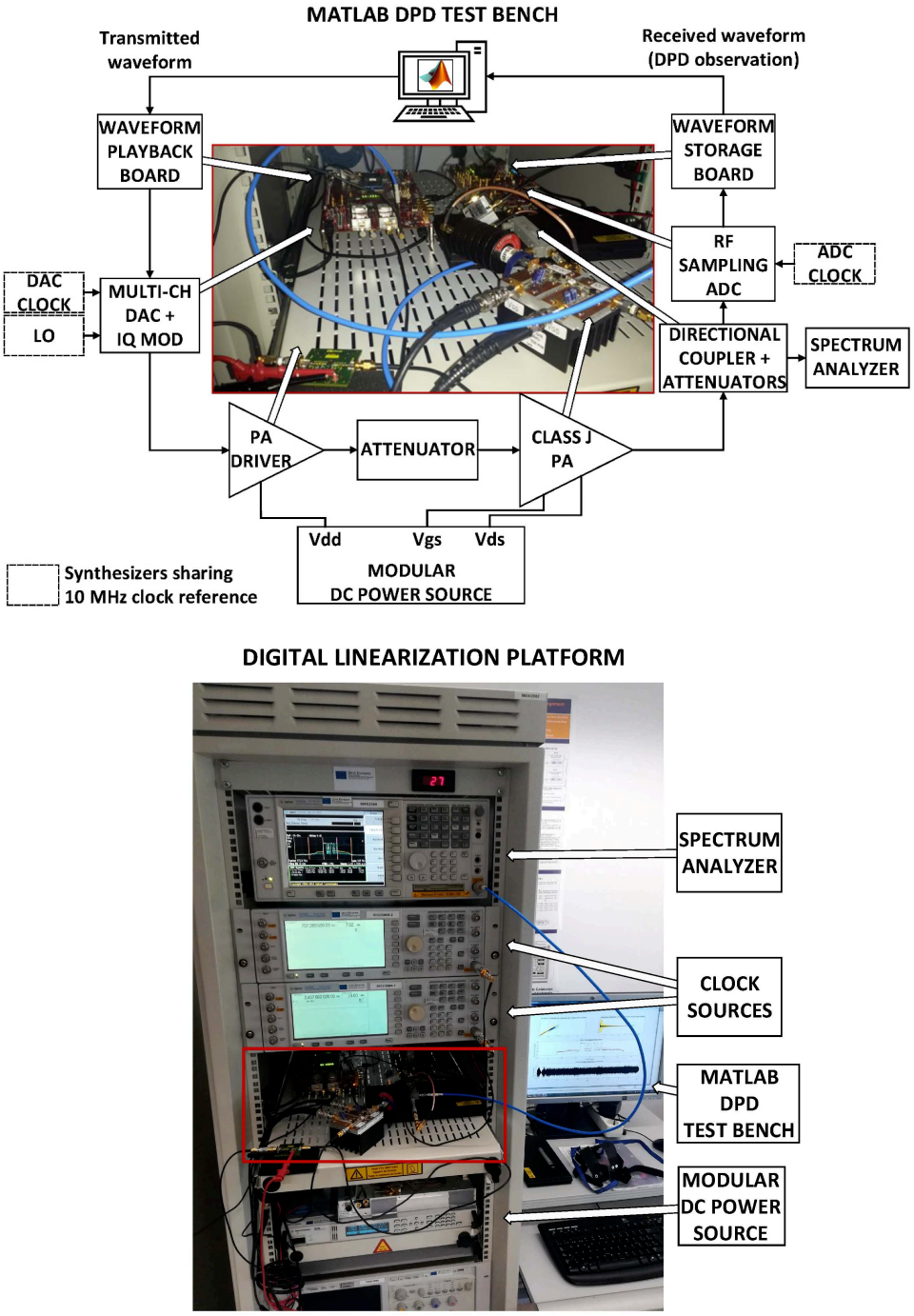


Figure 5.1: Block diagram of the DPD linearization test bench used for experimental validation including a picture of the PA used (upper). Picture of the overall Matlab-controlled digital linearization platform including the laboratory instrumentation used (below). Adapted from [67].

5.4.2 Experimental Results

First, the proposed BD-APCA technique is validated for PA behavioral modeling. The experiments have been conducted with different kinds of data, PAs and behavioral models:

- *Data 1*: 3×20 MHz LTE signals amplified by a Class-AB PA. The behavioral model used to model the PA is the MP (described in (3.2)).
- *Data 2*: 20 MHz FC-FBMC signals amplified by Class-J PA. The behavioral model used to model the PA is the GMP-LUT (described in (5.5)).
- *Data 3*: 80 MHz FC-FBMC signals amplified by Class-J PA. The behavioral model used to model the PA is the GMP (described in (3.3)).

The LUT-based behavioral model GMP-LUT is obtained by describing the non-linear functions of the GMP model in (3.3) by look-up tables instead of polynomials. The LUT-based models used in this dissertation follow the linear interpolation and extrapolation approach described in Chapter 3. Therefore, the GMP-LUT behavioral model can be defined as

$$\begin{aligned}\widehat{y}[n] = &\sum_{i=0}^{L_a-1}\sum_{p=0}^{P_a-1} x[n-\tau_i^a] f_{\overline{\Phi}_i}(|x[n-\tau_i^a]|)+ \\ &\sum_{i=0}^{L_b-1}\sum_{j=i-K_b}^{i+K_b}\sum_{p=1}^{P_b} x[n-\tau_i^b] f_{\overline{\Phi}_{i,j}}(|x[n-\tau_j^b]|)+ \\ &\sum_{i=0}^{L_c-1}\sum_{j=i-K_c}^{i+K_c}\sum_{p=1}^{P_c} x[n-\tau_i^c] f_{\overline{\Phi}_{i,j}}(|x[n-\tau_j^c]|)\end{aligned} \tag{5.5}$$

where $f_{\overline{\Phi}}(\cdot)$ is a is a piecewise linear complex function, defined in (3.7) as the linear combination of K basis functions.

The performance of the BD-APCA algorithm is compared to the performance of the full LS estimation in terms of $NMSE$, $ACEPR$, and the number of required coefficients. Note that when applying the BD-APCA solution, we consider a 10%

maximum degradation level allowed in either the $NMSE$ or the $ACEPR$ with respect to the values obtained with the full LS estimation.

Table 5.1 shows the comparison between the BD-APCA and the full LS for PA behavioral modeling. As observed in Table 5.1, even with different kinds of data and behavioral models, the BD-APCA technique reduces a huge number of coefficients (about $60 - 90\%$), while its modeling performance degradation (in terms of $NMSE$ and $ACEPR$) in comparison to the full LS (which includes a very high number of coefficients to reach the best performance) is below 10%. In the case of $Data$ 3, for example, the BD-APCA reduces up to 90% the number of coefficients, while losing only 2% of $NMSE$ and 9% of $ACEPR$ with respect to the full LS case.

Figure 5.2 presents the relationship between the number of basis functions (also the number of corresponding coefficients) of the PA model and the modeling accuracy in case of estimating the coefficients by means of the proposed BD-APCA solution. As it can be observed in Figure 5.2, in all three tests cases the modeling accuracy (in terms of $NMSE$ and $ACEPR$) converges as the number of basis functions of the PA model increases. Therefore, after the $ACEPR$ and $NMSE$ values reach a steady state and no significant improvement is observed, adding more coefficients is unnecessary and could even be harmful (i.e., drive to an ill-conditioned estimation).

The BD-APCA technique was also experimentally evaluated for DPD linearization using $Data3$ and the GMP behavioral model. Table 5.2 illustrates the comparison of DPD linearization performance in the following cases:

- without DPD,
- with DPD applying the full LS,
- with DPD applying the BD-APCA technique.

As shown in Table 5.2, the full DPD method that employs 322 coefficients found by LS regression has the best performance but poorest performance vs. hardware complexity trade-off. The BD-APCA DPD method employs 6.5 times less coefficients

Table 5.1: PA Behavioral Modeling Comparison.

Data	Behavioral model	Number of coefficients			$NMSE$ [dB]			$ACEPR$ [dB]		
		Full LS	BD-APCA	↓%	Full LS	BD-APCA	↓%	Full LS	BD-APCA	↓%
Data 1 : 3×20 MHz LTE (Class-AB PA)	MP	63	21	67	-37.3	-34.2	8	-44.0	-41.1	7
Data 2 : 20 MHz FC-FBMC (Class-J PA)	GMP-LUT	93	34	63	-43.5	-39.4	9	-49.7	-44.9	10
Data 3 : 80 MHz FC-FBMC (Class-J PA)	GMP	322	33	90	-41.8	-41.0	2	-51.4	-46.9	9

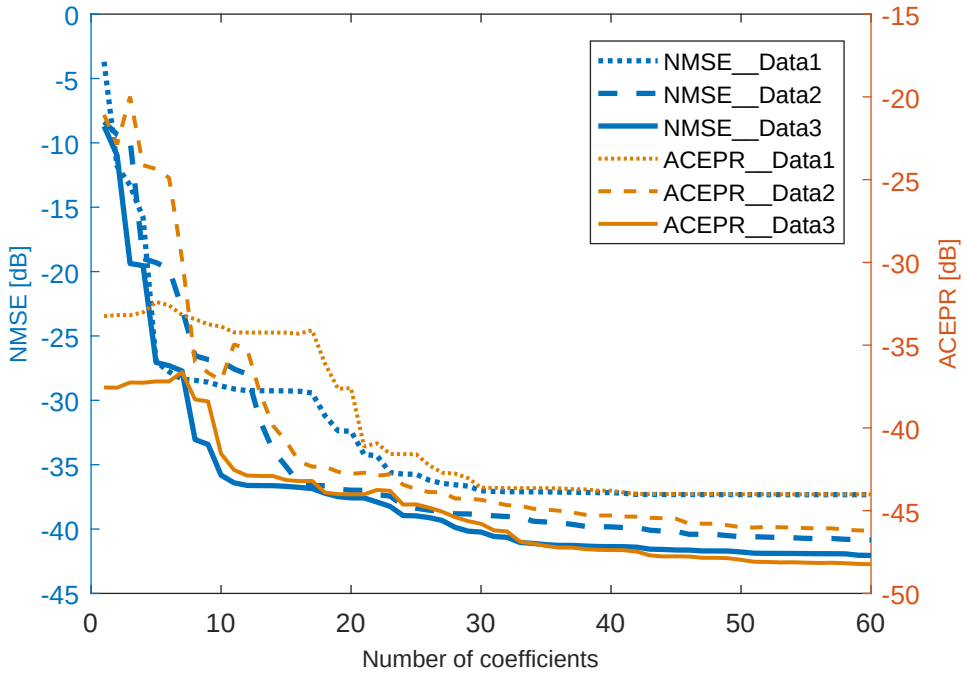


Figure 5.2: NMSE and ACEPR vs. number of components for three testing data sets.

Table 5.2: DPD Performance Comparison

Configuration 80 MHz FC-FBMC	Coeff No.	NMSE [dB]	$ACPR_{WCA}$ [dB]	EVM_{WCA} %
No DPD	-	-18.7	-36.5	5.7
Full DPD	322	-41.2	-50.5	1
BD-APCA DPD	50	-39.1	-45.6	1.3

than the full DPD approach, enables online implementation, performs similarly in terms of $NMSE$ and fulfills the -45 dB $ACPR$ requirement. Note that the amount of reduction (i.e., the minimum number of required coefficients) is conditioned by the fact that meeting the ACPR threshold specified in communications standards is a must (-45 dB in our particular case). The EVM improvement, thanks to DPD, facilitates using higher order subcarrier modulations such as 256-QAM. Figure 5.3 shows the unlinearized and linearized spectra for both the BD-APCA and full LS cases.

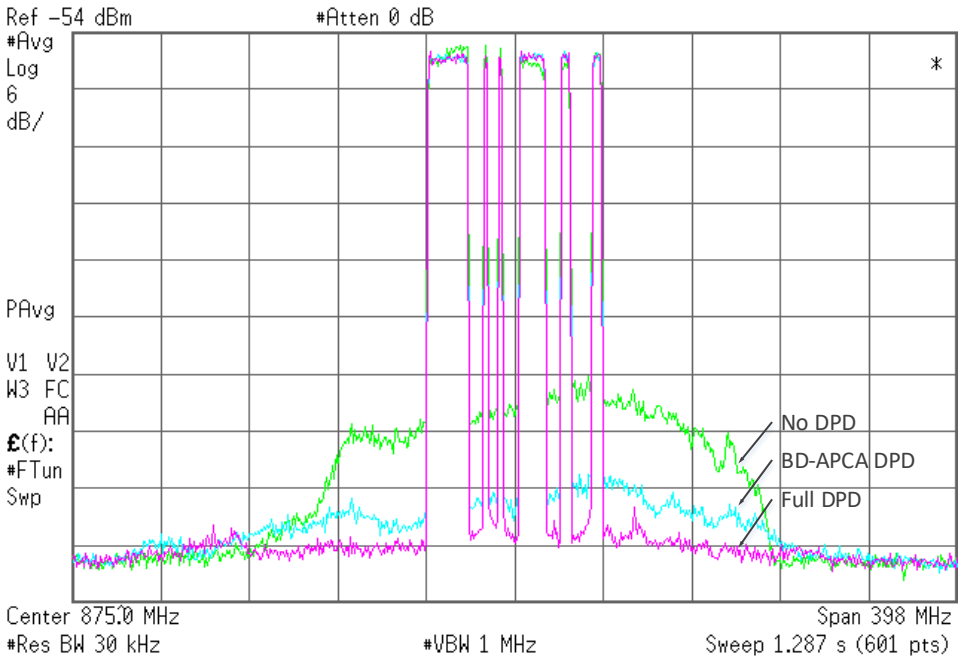


Figure 5.3: Unlinearized and linearized 80 MHz FC-FBMC output power spectra.

5.5 Discussion

The presented BD-APCA method enables an incremental updating scheme without the need of using the correlation matrix, which is advantageous over the conventional PCA since it allows BD-APCA to be implemented in an embedded processor. It also has advantages with respect to the conventional CGHA algorithm thanks to the proposed block-deflating scheme. The BD-APCA technique is compared to the full LS approach for the coefficient estimation and adaptation in the DPD feedback path. Unlike full LS, BD-APCA can be easily implemented online in an embedded processor to reduce the order of the coefficient estimation methods while still maintain high levels of linearization and behavioral modeling performance. Moreover, the coefficients can be estimated independently due to the orthogonality property of the new transformed basis. This idea is further developed by D. Lopez et al. in [67].

Chapter 6

Coefficient Estimation/Adaptation using Partial Least Squares and Dynamic Partial Least Squares

6.1 Introduction

In this Chapter, the PLS is employed as a method for estimating/adapting the coefficients in the DPD feedback path. Section 6.2 describes a DPD linearization approach for concurrent dual-band ET PAs. The multi-LUT DPD architecture, named 3-D distributed memory LUTs (3D-DML), which is converted from the MISO 3D-DMP model (presented in Subsection 3.1.2), is used to model the nonlinear behavior of the PAs. 3D-DML compensates for the distortion arising in concurrent dual-band ET PAs and is suitable for efficient FPGA implementation. In order to properly select the best LUTs of the 3D-DML model, a variant of the OMP algorithm, named OMP-LUT, is presented. OMP-LUT helps to reduce the number of required DPD coefficients in the forward path. Whereas in the feedback path, a technique estimating the coefficients of the 3D-DML architecture based on the PLS regression is suggested.

Section 6.2 provides the experimental results of DPD linearization for the concurrent dual-band ET PA employing the OMP-LUT in the DPD forward path and PLS technique in the DPD feedback path. The results prove how it is possible to reduce the DPD complexity (i.e. the number of coefficients) in both forward path and

feedback path while meeting the targeted linearity levels. Moreover, the performance of PLS is also compared to PCA's.

In Section 6.3, a DPD dynamic linearization approach is proposed. The approach employs the dynamic PLS technique inside the DPD adaptation loop to actively adjust the basis matrix in the DPD identification subsystem. The dynamic basis reduction is carried out at every iteration according to the residual linearization error. The dynamic PLS method gives similar linearization performance to the LS estimation solved via QR decomposition, but using less coefficients at every adaptation iteration. Section 6.3 also provides the experimental results of DPLS technique. The proposed dynamic linearization approach leads to a reduction of the number of estimated DPD coefficients (which impacts in the computational complexity reduction) and guarantees a well-conditioned and robust DPD coefficient estimation.

Finally, Section 6.4 gives some discussion about the PLS and DPLS techniques.

6.2 PLS Identification of Multi Look-Up Table Digital Predistorters for Concurrent Dual-Band ET PAs

6.2.1 Forward DPD path

3D-DML Digital Predistorter

The MISO DPD model 3D-DMP in (3.4) is proved to be a good choice for describing the nonlinearity of concurrent dual-band ET PAs [11]. In order to simplify the DPD for DB ET PAs and targeting a FPGA implementation, the 3D-DML DPD model is introduced. The 3D-DML is based on the 3D-DMP model. The polynomials in the 3D-DMP are converted to LUTs with linear/bilinear interpolation and extrapolation.

The 3D-DML DPD model for the signal in Band 1 is defined as:

$$\begin{aligned} x_1[n] &= \sum_{i=0}^{N_1-1} u_1[n-\tau_i^{u_1}] f_{\overline{\Phi}_{1,i}}\Big(\big|u_1[n-\tau_i^{u_1}]\big|\Big) \\ &+ \sum_{i=1}^{N_2-1}\sum_{j=1}^{M_2-1} u_1[n] f_{\overline{\Phi}_{1,i,j}}\Big(\big|u_1[n-\tau_i^{u_1}]\big|, \big|u_2[n-\tau_j^{u_2}]\big|\Big) \\ &+ \sum_{i=1}^{N_3-1}\sum_{k=1}^{K_3-1} u_1[n] f_{\overline{\Phi}_{1,i,k}}\Big(\big|u_1[n-\tau_i^{u_1}]\big|, E[n-\tau_k^{e}]\Big) \end{aligned} \tag{6.1}$$

where N_1, N_2 and N_3 are the numbers of delays of the input signal $u_1[n]$ at each branch; M_2 is the number of delays of the interference signal $u_2[n]$; K_3 is the number of delays of the supply envelope $E[n]$; τ^{u_1}, τ^{u_2} and τ^{e} (with $\tau^{u_1,u_2,e} \in \mathbf{Z}$ and $\tau_0^{u_1,u_2,e} = 0$) are the most significant sparse delays of the input ($u_1[n]$), interference signal ($u_2[n]$) and envelope ($E[n]$). Moreover, following the LUT interpolation and extrapolation concept in [57], $f_{\overline{\Phi}_{1,i}}(u_1)$ in (6.1) represents a 1-D LUT and is a piecewise linear complex function defined as the linear combination of N basis functions; while $f_{\overline{\Phi}_{1,i,j}}(u_1, u_2)$ or $f_{\overline{\Phi}_{1,i,k}}(u_1, E)$ in (6.1) are 2-D LUTs defined by a piecewise bilinear complex function. Further details on the bilinear interpolation and extrapolation can be found in [57, 99].

Analogously, the DPD function for Band 2 can be defined as in (6.1) but with $u_2[n]$ and $u_1[n]$ being the input and interfering signal, respectively.

Best LUTs Selection Method (OMP-LUT)

In the 3D-DML DPD model, the required number of coefficients to compensate for the in-band, cross-band intermodulation distortion and the slow-envelope dependent distortion in ET PAs is seriously high. This leads to an increase of the system's computational complexity. Besides, the sparse data of LUT-based DPD model drives the system to overfitting and uncertainty.

The sparsity of the LUT-based DPD models can be exploited to reduce the number

of required basis functions or active components by using the greedy methods (such as matching pursuit and OMP) or regression methods (such as LASSO, Ridge and Elastic Net). By using OMP, for example, it is possible to obtain a sorted set of the most relevant basis functions, named the OMP list. Moreover, the Bayesian information criterion can be applied to OMP to determine the most suitable number of basis functions.

Since 3D-DML model is built in LUTs, direct applying OMP to the 3D-DML data matrix is impractical. Thus, to retain the effect of LUTs on the 3D-DML model, we propose a method to allow doing the selection in LUTs, instead of the individual basis functions (i.e. the columns of the 3D-DML data matrix). The proposed method OMP-LUT is described in the following.

In order to value the significance of a LUT i on the 3D-DML model, the times a LUT i appears in the OMP list (t_i) is considered

$$t_i = \sum_{j=1}^{n} a_j \tag{6.2}$$

in which n is the number of basis functions (columns) of the LUT i; if column j of LUT i is in the OMP list $a_j = 1$, otherwise $a_j = 0$.

It is also needed to *a*) discriminate between LUTs which have same size n and appearance times t in the OMP list; and *b*) determine a criterion to make different-size LUTs comparable. The OMP algorithm sorts the basis functions by relevance (i.e., the later a basis function appears in the list, the less significant it is). Therefore, to enable *a*), a weight w is given to each element in the OMP list. w is an assigned small positive value. For the first element in the list, its weight is the smallest one. In the experiment in [99], the initial value w is assigned to 0.01; and then this value is increased every position by 0.01. The sum of weight of a LUT i is

$$s_i = \sum_{j=1}^{n} w_j \tag{6.3}$$

in which w_j is the weight of the column j in the LUT i. Given the LUT i_1 and LUT i_2 with $t_{i_1} = t_{i_2}$, if $s_{i_1} < s_{i_2}$ then the LUT i_1 is considered to be more relevant than the LUT i_2, and vice versa.

To solve b), the final weight (l) of a LUT i is computed as follow $l_i = (t_i - s_i)/n$. The greater l is, the better the LUT is. The list of LUTs is ranked in decreasing order according to the value l of each LUT. A small number of LUTs used as starting point is increased until meeting linearity requirements.

6.2.2 Feedback Identification/Adaptation Path

In order to avoid the ill-conditioned LS in (3.38), several regularization techniques [100] can be found in literature to address this problem. Among them, the PLS regression method is a suitable technique since it not only enhances the conditioning of the estimation but also allows reducing the number of coefficients to be estimated. The process of employing PLS regression in DPD feedback path can be referred in Subsection 4.2.2.

As presented in Subsection 4.2.2, both PLS and PCA construct new components that are the linear combinations of the original basis functions. However, while PCA obtains new components that maximize their own variance, PLS finds linear combinations of the original basis functions that maximize the covariance between the new components and the reference signal (i.e. the output vector in PA modeling or the error vector in DPD adaptation). With that, PLS improves the accuracy of the DPD coefficient estimation. Consequently, PLS's performance is better than PCA's.

6.2.3 Experimental Results

Experimental Test Bench

The DPD estimation/adaptation methods using PLS and PCA applied on the D3-DML behavioral model [99] are tested with the remoteUPCLab test bench (see Figure

6.1).

The test bed is built from the collaboration between our lab with IEEE MTT-S and Rohde & Schwarz (R&S). It consists of a PC running Matlab and an FTP server to allow worldwide users to connect to the equipments:

- the R&S SMW200A vector signal generator,
- the R&S FSW8 signal and spectrum analyzer,
- the DUT consisting in a Texas Instruments LM3290-91-1EVM ET board that includes a Skyworks SKY776621 4G handset PA. The PA operates with a dual-band signal composed by two OFDM signals whose center frequency is spaced 80 MHz and that feature 10 MHz and 5 MHz bandwidth (with 80 MHz spacing).

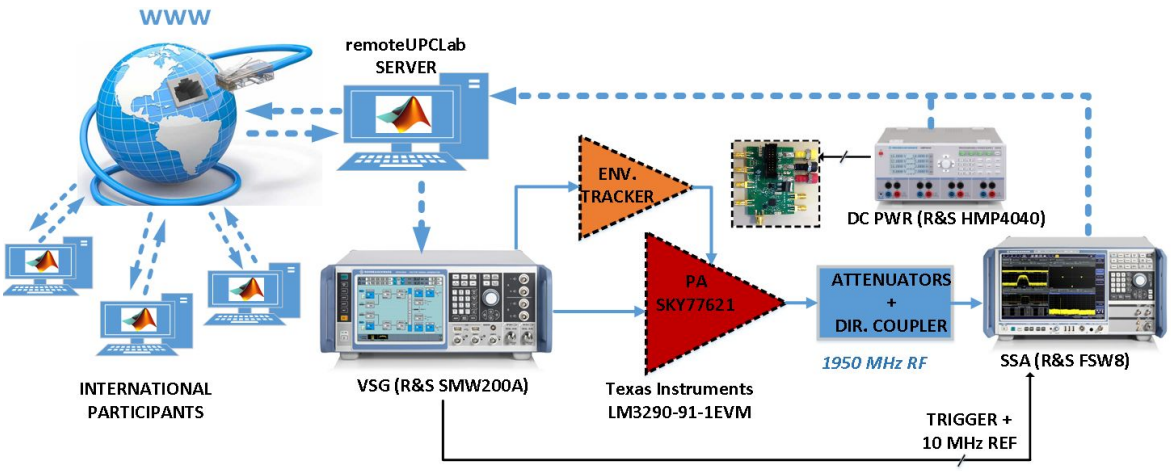


Figure 6.1: Block diagram of the remoteUPCLab.

The remoteUPCLab server receives the incoming baseband I/Q waveforms and an appropriate delay-compensated supply waveform from a remote user. These are both downloaded into the R&S SMW200A vector signal generator (VSG) that generates *i*) the I/Q signals being RF up converted to deliver the PA input signal (i.e. through the VSG I/Q modulator) and *ii*) the EVM supply modulator input signal. The R&S FSW8 signal and spectrum analyzer (SSA) is in charge of RF down conversion and

data acquisition of the waveform at the output of the PA, whose I/Q data will be sent back to the remote user for DPD processing. The baseband clock that is employed in the signal processing operations is 122.88 MHz which corresponds also to the I/Q A/D and D/A sampling frequencies (no over-sampling is applied). The peak output power level from the SKY77621 PA is limited to approximately 1 W. The settings in the signal analyzer (reference level and input attenuation) are set in such a way as not to distort the measured signal even for the highest peak power levels allowed.

Experimental Results

In DPD forward path, three different selection methods are applied to the original data matrix (composed of 223 basis functions) to show the model order reduction capabilities while meeting the linearity specifications (set at -45 dB of $ACPR$). They are:

M1) No OMP: no proper search is carried out, the look-up table selection is done by adding consecutive memory terms of both input signals and the slow envelope. It is likely that with this straightforward method to build the data matrix (no requirement or constraint is applied to select LUTs), some basis functions will be highly correlated among them.

M2) OMP-col: selection of the best columns of the data matrix using the OMP-BIC algorithm [77]. With this method, the basis functions are selected without taking into account to which LUT they belong to, thus, without concerning if the resulting selected basis correspond to complete LUTs or not. The output of the search are the best columns of the data matrix. With the OMP-col, we are implicitly avoiding the ill-conditioning problem by selecting only the most relevant basis functions.

M3) OMP-LUT: selection of the best LUTs using the modified version of the OMP-BIC algorithm. The output of the search are the best complete LUTs of the data

matrix. For the sake of completeness some basis functions with low relevance have to be added to complete the LUT structure and this may later contribute to introduce uncertainty in the estimation of the coefficients.

The linearity, the power efficiency and the reduced numbers of basis functions of the DPD model after applying three aforementioned selection methods are shown in Table 6.1. These results are taken when the system meets the $ACPR$ threshold of -45 dB. Applying the method M1, which mainly consist in adding delays sequentially, may lead to an ill-conditioned estimation and therefore require a large number of coefficients to achieve the threshold of -45 dB of $ACPR$. Note the DPD for Band 2 for example, requiring up to 153 coefficients or, what is the same, 2 1-D LUTs and 10 2-D LUTs. When selecting the most relevant coefficients with OMP, the original universe of possible coefficients is limited to 110 coefficients for Band 1 and 119 for Band 2. As expected, thanks to the OMP algorithm, both methods M2 and M3, allow reaching the targeted $ACPR$ value in both bands with less coefficients than method M1. However, while the OMP-col method requires 92 coefficients for both bands to get -45 dB of $ACPR$, the proposed OMP-LUT method M3, allows meeting the requirements with only 62 coefficients (i.e., a 3D-DML DPD with 2 1-D LUTs and 5 2-D LUTs) for Band 1 and 73 coefficients (i.e., a 3D-DML DPD with 3 1-D LUTs and 4 2-D LUTs) for Band 2.

Then, starting from the reduced set of coefficients obtained for the DPD function in the forward path, we now attempt to improve the conditioning of the estimation as well as reducing the number of coefficients to be estimated in the feedback identification/adaptation path. For simplicity (without loss of generality), the advantages of the PLS-based estimation will be highlighted considering the extraction of the coefficients of the 3D-DML PA behavioral model instead of the 3D-DML DPD model.

In order to extract the coefficients of the DPD function (based on LUTs) in the forward path, the Moore-Penrose inverse is commonly used to solve the LS regression. However, if the resulting order reduced matrix is not well-conditioned, then the

Table 6.1: Comparison of Different OMP Coefficient Selection Methods for 3D-DML DPD

Method	Pout [dBm]	η [%]	$NMSE$ [dB]	$ACPR$ [dB]	No. coeff.
(a)			B1: -36.5	B1: -45.6	B1: 85
No OMP	23.1	19.0	B2: -37.7	B2: -46.0	B2: 153
(b)			B1: -36.3	B1: -45.1	B1: 92
OMP-col	22.8	18.2	B2: -37.5	B2: -45.3	B2: 92
(c)			B1: -36.5	B1: -45.1	B1: 62
OMP-LUT	23.0	18.7	B2: -37.1	B2: -45.3	B2: 73

coefficient estimation may lead to an inaccurate solution. In the particular case of the OMP-col selection, since all the basis functions were properly selected, the LS estimation will be perfectly conditioned. However, in the OMP-LUT case, because the objective is to obtain an integer number of complete LUTs we have to include some of the basis functions (i.e., columns) that make the LS estimation rank deficient. To illustrate this, Table 6.2 shows the results of PA behavioral modeling in terms of $NMSE$ and $ACEPR$ for the three selection methods when considering the Moore-Penrose inverse and the Matlab's backslash (\) operations. It can be observed that in the case of the OMP-LUT basis selection, the MP-LS cannot provide an accurate estimation, while the Matlab's backslash operator can.

As an alternative to the backslash operator, we suggest to use the PLS algorithm to both improve the conditioning of the correlation matrix and reduce the number of coefficients of the estimation. In addition, we also compare the accuracy versus coefficient reduction between the PLS and PCA techniques. The Figures 6.2-6.4 show the $NMSE$ and $ACEPR$ evolution when considering different numbers of coefficients in the identification for three basis function selection methods: M1, M2 and M3, respectively. In all three cases, we can see that the PLS technique is more robust than PCA in terms of $NMSE$ and $ACEPR$ degradation when reducing the number

Table 6.2: Moore-Penrose inverse vs. Matlab's Backslash Operator for the PA Behavioral Modeling after Applying Different Coefficient Selection Methods

Method	Sig.	No. of coeff.	$NMSE$ [dB]		$ACEPR$ [dB]	
			MP–LS	\	MP–LS	\
(a)	B1	85	NaN	-39.8	NaN	-43.8
No OMP	B2	153	NaN	-46.0	NaN	-51.1
(b)	B1	92	-39.4	-39.4	-43.4	-43.4
OMP-col	B2	92	-45.9	-45.9	-50.7	-50.7
(c)	B1	62	NaN	-40.0	NaN	-44.0
OMP-LUT	B2	73	NaN	-45.6	NaN	-50.3

of coefficients of the estimation (coefficients of the transformed basis). The reason for this is that PLS, unlike PCA, takes also into account the information of the output signal for creating the transformation matrix.

In case of M1 (no OMP), in Figure 6.2, there are two types of $NMSE$ and $ACEPR$ degradation, when the number of identification components is too small and also when the full basis of components is considered. The latter degradation is due to the fact that the correlation matrix is ill-conditioned when considering the full basis of new components because we are including the ones that are expendable. Eliminating the less relevant columns (components) produces a regularization effect, which results in a new well-conditioned basis with less coefficients to estimate. Similarly, in case of M3 (OMP-LUT) in Figure 6.4 we can observe the same behavior involving $NMSE$ and $ACEPR$ degradation due to excess of coefficient reduction or due to the ill-conditioned estimation for an excess of linear dependent components. Instead, in case of M2 (OMP-col) in Figure 6.3, thanks to the proper basis selection performed by the OMP-BIC algorithm, no ill-condition problem is observed at high number of components, while only degradation is appreciated, as expected, when significantly decreasing the number of coefficients.

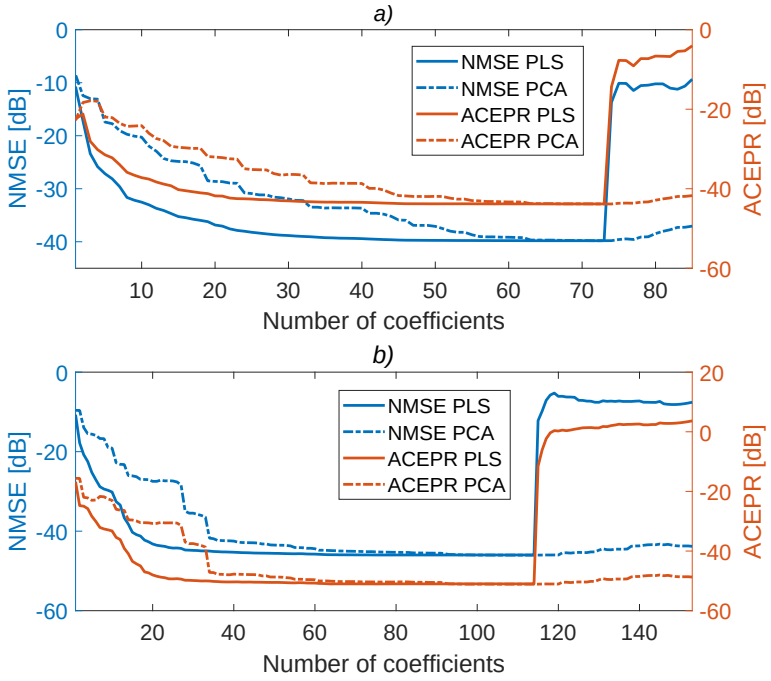


Figure 6.2: $NMSE$ and $ACEPR$ vs. number of coefficients considering No-OMP selection in the forward path, *a)* for Band 1 signal and *b)* for Band 2 signal.

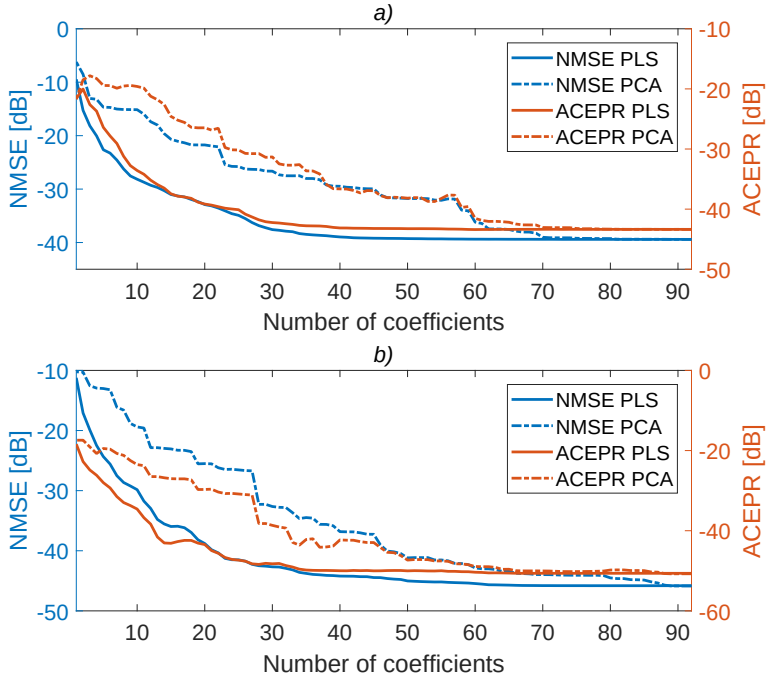


Figure 6.3: $NMSE$ and $ACEPR$ vs. number of coefficients considering OMP-col selection in the forward path, *a)* for Band 1 signal and *b)* for Band 2 signal.

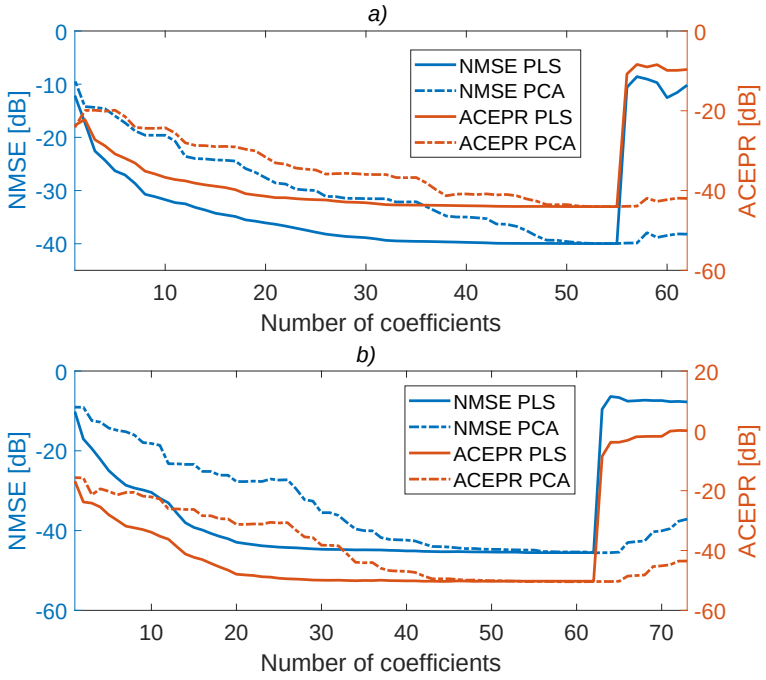


Figure 6.4: $NMSE$ and $ACEPR$ vs. number of coefficients considering OMP-LUT selection in the forward path, *a)* for Band 1 signal and *b)* for Band 2 signal.

The advantage of using the PLS technique for the coefficients estimation when considering the case of the OMP-LUT basis selection is summarized in Table 6.3. Thanks to PLS, we can reduce the number of coefficients to be estimated almost without losing accuracy in the identification. In particular (see Table 6.3), around 35% for Band 1 and 34% for Band 2 of reduction in the number of coefficients can be considered at the expenses of a loss of identification performance (in terms of $NMSE$ and $ACEPR$) of less than 0.5% in the worst case.

Figure 6.5 (for Band 1) and Figure 6.6 (for Band 2) show the spectra of the estimated outputs considering a 3D-DML behavioral model after applying OMP-LUT selection (i.e., 62 coefficient Band 1 and 73 coefficient Band 2) and reduced PLS coefficient estimation (i.e., 40 coefficient Band 1 and 48 coefficient Band 2). Finally, applying the aforementioned combination (i.e., OMP-LUT selection in the forward path and PLS reduction estimation in the feedback identification path) we can significantly reduce the complexity of the 3D-DML DPD while still meeting the

Table 6.3: PLS-based Behavioral Modeling Identification after OMP-LUT coefficient reduction

Signal	Number of coefficients			$NMSE$ [dB]			$ACEPR$ [dB]		
	"\"	PLS reduction	↓ %	"\"	PLS reduction	↓ %	"\"	PLS reduction	↓ %
Band 1	62	40	35.5	-40.0	-39.8	0.5	-44.0	-43.8	0.5
Band 2	73	48	34.2	-45.6	-45.4	0.4	-50.3	-50.3	0.0

specific linearity requirements as shown in Figure 6.7.

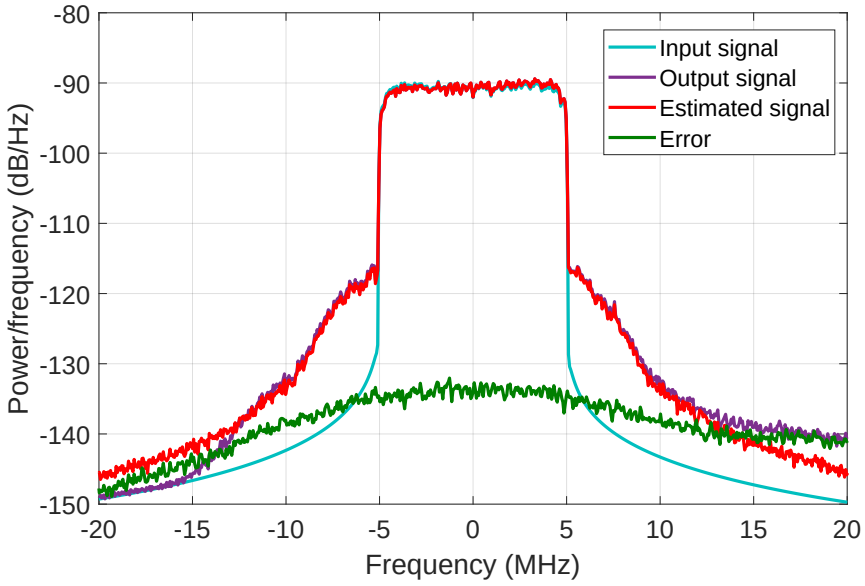


Figure 6.5: Spectra of the input, measured output, estimated output and residual error of Band 1 signal when applying 'OMP-LUT and PLS reduction'.

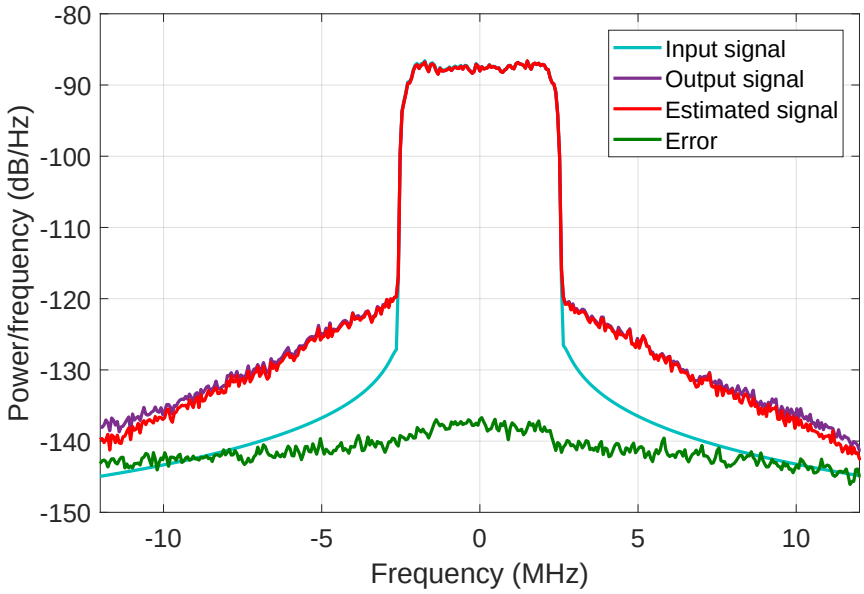


Figure 6.6: Spectra of the input, measured output, estimated output and residual error of Band 2 signal when applying 'OMP-LUT and PLS reduction'.

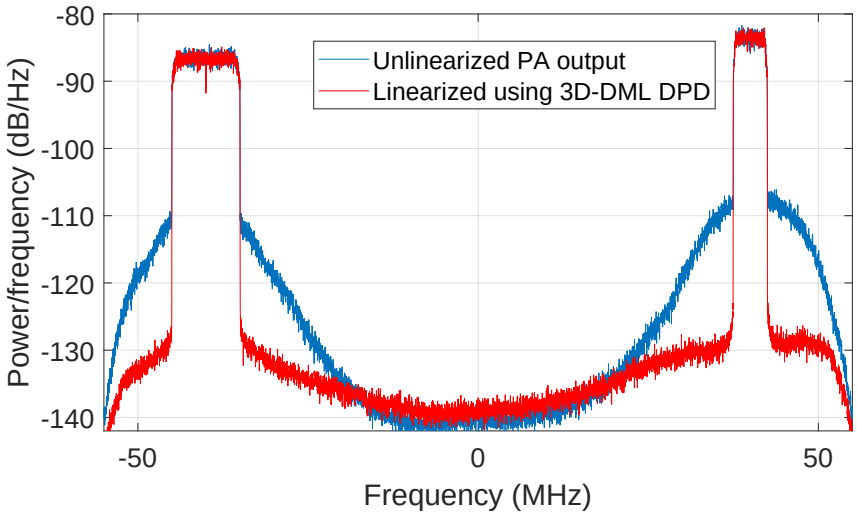


Figure 6.7: Unlinearized and linearized (using the 3D-DML DPD and the OMP-LUT configuration in Table 6.1) spectra of a dual-band signal.

6.3 Dynamic Selection and Update of Digital Predistorter Coefficients for Power Amplifier Linearization

6.3.1 Dynamic Partial Least Squares algorithm

In a closed-loop direct learning approach (see Figure 3.4), the DPD coefficients are iteratively updated as described in (3.37). In order to reduce the number of coefficients in the DPD forward path, OMP is applied to choose the most relevant basis functions of $\boldsymbol{U}$. Then, in order to avoid calculating the Moore-Penrose inverse of the correlation matrix of $\boldsymbol{U}$, the PLS regression [101] is implemented inside the DPD adaptation loop to *dynamically* adjust the basis matrix in the DPD subsystem. This solution is an alternative of QR-LS.

The dynamic orthonormal transformation matrix (DOTM) algorithm (Algorithm 4) is proposed to generate a new transformed matrix with a minimum necessary number of new components. DOTM algorithm is a modification of the iterative SIMPLS algorithm [95] for PLS regression. Whereas the size of the transformation matrix $\boldsymbol{P}_{dpls}$ is predetermined and given as an input information in SIMPLS, in DOTM, the number of columns of $\boldsymbol{P}_{dpls}$ and coefficients are iteratively calculated and added until the power of the estimated error is close enough to the threshold E_{th}. E_{th} is defined as a percentage δ of the power of the error signal $\boldsymbol{e}$.

Figure 6.8 depicts the proposed DPD estimation/adaptation employing dynamic basis matrix approach. Thanks to the DOTM algorithm, at each iteration i of the DPD adaptation, the number of most important columns of the transformation matrix $\boldsymbol{P}_{dpls}$ varies, and thus only the minimum necessary number of columns that meets the E_{th} threshold requirements is selected.

The $M \times L$ transformation matrix $\boldsymbol{P}_{dpls}$ obtained from DOTM is used for constructing the transformed basis matrix $\widehat{\boldsymbol{U}}$ as depicted in (4.16), in which $\boldsymbol{U}$ being the

$N \times M$ matrix of basis functions after applying OMP. The transformed matrix $\widehat{\boldsymbol{U}}$ presents orthonormal components and has dimensions $N \times L$.

Note that the number of components L ($L \leq M$) is variable since it depends on the dimensions of $\boldsymbol{P}_{dpls}$. The new orthonormal components (i.e., columns) of the transformed matrix $\widehat{\boldsymbol{U}}$ are sorted according to their contribution to maximize the covariance between the new components and the error signal $\boldsymbol{e}$.

Taking into account the orthonormal property of the transformed matrix $\widehat{\boldsymbol{U}}$ (i.e., $\widehat{\boldsymbol{U}}^H\widehat{\boldsymbol{U}} = \boldsymbol{I}$), the update of the transformed coefficients in (4.17) is simplified as

$$\boldsymbol{\Delta}\widehat{\boldsymbol{w}} = \mu\widehat{\boldsymbol{U}}^H\boldsymbol{e}. \tag{6.4}$$

Then, the increment of the original coefficients is obtained through the following anti-transformation as shown in (4.20).

Since the components of $\widehat{\boldsymbol{U}}$ are orthonormal, the power of the estimated error ($\boldsymbol{e}^H\boldsymbol{e}$) is equivalent to the power of the increment of estimated coefficients ($\Delta\widehat{\boldsymbol{w}}^H\Delta\widehat{\boldsymbol{w}}$). Following is the demonstration for that. It can be deduced from (3.38) that

$$\boldsymbol{e} = \boldsymbol{U}\Delta\boldsymbol{w}. \tag{6.5}$$

Similarly, with $\Delta\widehat{\boldsymbol{w}}$ and $\widehat{\boldsymbol{U}}$, we have

$$\boldsymbol{e} = \widehat{\boldsymbol{U}}\Delta\widehat{\boldsymbol{w}}. \tag{6.6}$$

Therefore,

$$\boldsymbol{e}^H\boldsymbol{e} = (\widehat{\boldsymbol{U}}\Delta\widehat{\boldsymbol{w}})^H\widehat{\boldsymbol{U}}\Delta\widehat{\boldsymbol{w}} = \Delta\widehat{\boldsymbol{w}}^H\widehat{\boldsymbol{U}}^H\widehat{\boldsymbol{U}}\Delta\widehat{\boldsymbol{w}} = \Delta\widehat{\boldsymbol{w}}^H\Delta\widehat{\boldsymbol{w}}. \tag{6.7}$$

The power of the increment of estimated coefficients $\Delta\widehat{\boldsymbol{w}}^H\Delta\widehat{\boldsymbol{w}}$ is used to control the while loop in Algorithm 4.

As described in Figure 6.8, the DPD adaptation will continue until achieving the

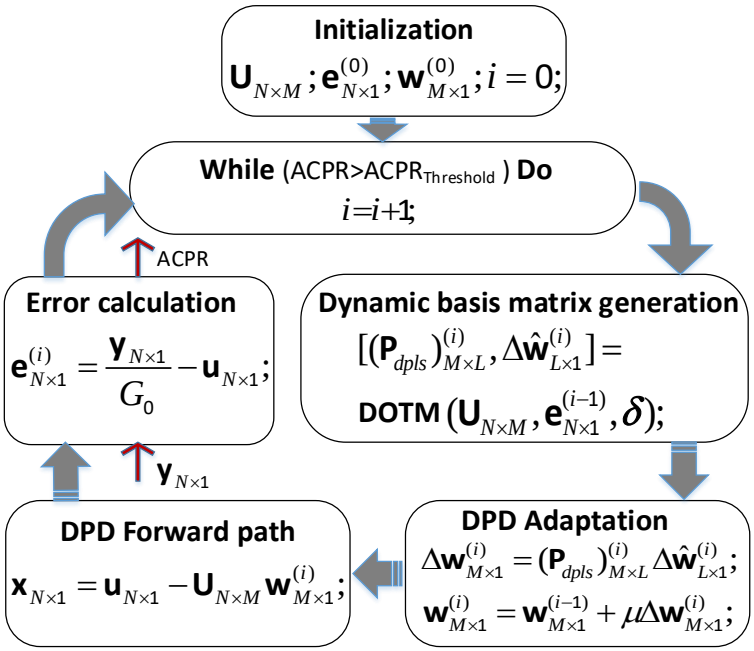


Figure 6.8: Flowchart of the DPD estimation/adaptation using dynamic basis matrix.

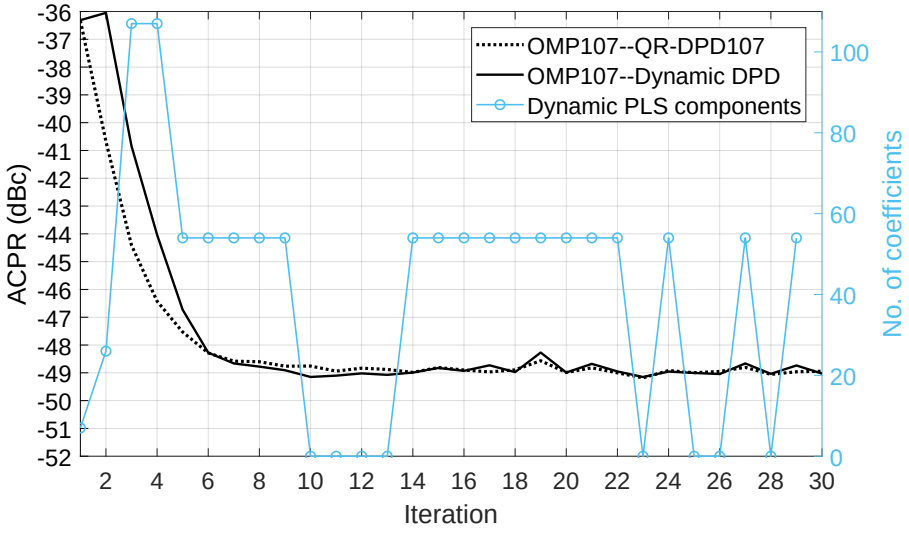


Figure 6.9: The DPD estimation/adaptation with DOTM vs. with QR decomposition.

desired $ACPR$ level.

Algorithm 4 DOTM Calculation

procedure DOTM($\boldsymbol{U}, \boldsymbol{e}, \delta$)
  initialization:
  $\boldsymbol{V}^{(0)} = \{\}; \quad \boldsymbol{P}_{dpls}^{(0)} = \{\}; \quad \Delta\widehat{\boldsymbol{w}}^{(0)} = \{\}; \quad j = 1;$
  $E_1 = 0; \quad E_{th} = \delta\mu^2||\boldsymbol{e}||_2^2; \quad \boldsymbol{r}_1 = \boldsymbol{U}^H\boldsymbol{e};$
  while $E_j < E_{th}$ **do**
    $\boldsymbol{p}_j = \dfrac{\boldsymbol{r}_j}{||\boldsymbol{U}\boldsymbol{r}_j||_2};$
    $\boldsymbol{P}_{dpls}^{(j)} \leftarrow \boldsymbol{P}_{dpls}^{(j-1)} \bigcup \boldsymbol{p}_j;$
    $\boldsymbol{v}_j = \boldsymbol{U}^H\boldsymbol{U}\boldsymbol{p}_j;$
    for $repeat = 1$ to 2 **do**
      for $i = 1$ to $j - 1$ **do**
        $\boldsymbol{v}_i = \boldsymbol{V}(:, i);$
        $\boldsymbol{v}_j = \boldsymbol{v}_j - (\boldsymbol{v}_i^H\boldsymbol{v}_j)\boldsymbol{v}_i;$
      end for
    end for
    $\boldsymbol{v}_j = \dfrac{\boldsymbol{v}_j}{||\boldsymbol{v}_j||_2};$
    $\boldsymbol{r}_j = \boldsymbol{r}_j - \boldsymbol{v}_j(\boldsymbol{v}_j^H\boldsymbol{r}_j);$
    $\boldsymbol{V}^{(j)} \leftarrow \boldsymbol{V}^{(j-1)} \bigcup \boldsymbol{v}_j;$
    $\boldsymbol{r}_{j+1} = \boldsymbol{r}_j - \boldsymbol{V}^{(j)}((\boldsymbol{V}^{(j)})^H\boldsymbol{r}_j);$
    $\widehat{u}_j = \boldsymbol{U}\boldsymbol{p}_j;$
    $d\widehat{w}_j = \widehat{\boldsymbol{u}}_j^H\boldsymbol{e};$
    $\Delta\widehat{\boldsymbol{w}}^{(j)} \leftarrow \Delta\widehat{\boldsymbol{w}}^{(j-1)} \bigcup d\widehat{w}_j;$
    $E_{j+1} = ||\Delta\widehat{\boldsymbol{w}}^{(j)}||_2^2;$
    $j = j + 1;$
  end while
  Return $\boldsymbol{P}_{dpls}, \Delta\widehat{\boldsymbol{w}}$
end procedure

6.3.2 Experimental Results

The dynamic selection and update of the DPD coefficients was validated using the Matlab-controlled digital linearization test bed described in 5.4.1. The original basis functions were generated by using the GMP behavioral model with 322 coefficients. By applying OMP in the DPD forward path, the number of coefficients is reduced down to 107. Then, we performed two tests: *i*) DPD estimation/adaptation with QR decomposition using 107 coefficients and *ii*) DPD estimation/adaptation using the

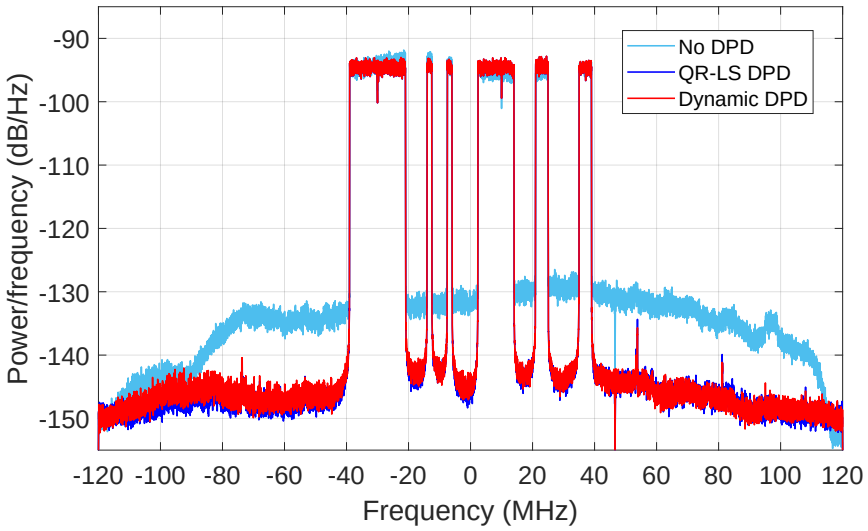


Figure 6.10: Spectra of the PA output before and after dynamic DPD linearization.

DOTM algorithm to generate dynamic basis matrix with an $ACPR$ target of -48 dB.

As shown in Figure 6.9, both DPD adaptation methods converge after 10 iterations to the same $ACPR$ level. However, while the DPD adaptation with QR-LS requires a constant number of 107 coefficients at every iteration to converge, the DPD adaptation with the DOTM algorithm (that dynamically adjusts the dimensions of the basis matrix), needs less coefficients (only at iterations 3 and 4 it needs up to 107 coefficients) to achieve the same targeted $ACPR$ value.

Figure 6.10 shows the non-linearized and linearized spectra considering the QR-LS and the proposed dynamic selection and update of the DPD coefficients. Despite showing similar performance, when comparing the measured running time for both algorithms implemented in Matlab plain code, the DOTM algorithm is around 50 times faster than the case of QR.

6.4 Discussion

In this Chapter, the PLS regression method is suggested to address the ill-conditioning problem of the DPD coefficient estimation. In order to validate the proposed PLS

method, the 3D-DML DPD model which is built from multiple LUTs is considered. PLS addresses both the regularization problem of the DPD coefficient estimation, and at the same time, reducing the number of coefficients to be estimated in the DPD feedback identification path. In addition, by exploiting the orthogonality of the PLS transformed matrix, the computational complexity of the coefficients' identification is remarkably simplified. Moreover, the PLS approach is compared to the PCA. The experimental results prove that PLS has over-performance to the PCA since PLS considers the reference signal when generating the new components whereas PCA does not.

Furthermore, a new method for dynamically estimating and updating the DPD coefficients using the DOTM algorithm, based on the PLS algorithm, is proposed. The DPLS is employed inside the DPD adaptation loop, allowing to decide the minimum number of coefficients required to minimize the residual error. The DPLS benefits from reducing the number of coefficients to be estimated in the DPD identification subsystem with respect to the non-dynamic conventional DPD estimation based on QR-LS.

Chapter 7

Combination of Principal Component Analysis and Dynamic Partial Least Squares

7.1 Introduction

This Chapter presents a new technique that dynamically estimates and updates the coefficients of a digital predistorter. The proposed technique is a combination of PCA and DPLS. The technique is dynamic in the sense that at every iteration of the coefficient's update, only a minimum necessary coefficients are estimated and updated according to a criterion based on the residual estimation error.

In the first step, the original basis functions defining the digital predistorter in the forward path are orthonormalized for DPD adaptation in the feedback path by means of a pre-calculated PCA transformation and normalization. The robustness and reliability of the pre-calculated PCA transformation is tested and verified. Note that the PCA transformation matrix is obtained off-line and only once.

In the second step, the dynamic PLS algorithm explained in Subsection 6.3.1, is applied to obtain the minimum number of most relevant transformed components required for updating the coefficients of the DPD linearizer. The combination of the PCA transformation with the DPLS extraction of components is equivalent to using the canonical correlation analysis (CCA) technique, which is optimum in the sense of

generating components with maximum correlation (instead of maximum covariance as in the case of the DPLS extraction alone). The relationship between PLS and CCA is analyzed in Subsection 7.2. Subsection 7.3 presents the method of combining PCA and DPLS for DPD estimation/adaptation.

The proposed dynamic extraction technique is evaluated and compared in terms of computational cost and linearization performance with the commonly used QR decomposition approach for solving the least squares problem. Experimental results in Subsection 7.4 show that the combination of PCA and DPLS drastically reduces the amount of DPD coefficients to be estimated while maintaining the same achieved linearity levels. Finally, Subsection 7.5 summarizes and discusses the proposed identification method consisting in the combination of PCA and DPLS.

7.2 Relationship between CCA and PLS

As presented in Subsection 4.1.2, CCA and PLS are techniques for feature extraction from a set of variables or basis functions. However, while CCA finds the components that have maximum *correlation*, PLS finds the components that have maximum *covariance* between the components and the reference signal. *Covariance* and *correlation* are two different statistical measures for quantifying how the variables are related. CCA creates new components that maximize the *correlation* factor ρ_{CCA}, defined as

$$\rho_{CCA} = \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{U}\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2}. \tag{7.1}$$

PLS maximizes the *covariance* factor ρ_{PLS}, defined as

$$\rho_{PLS} = \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} \tag{7.2}$$

with $\langle \cdot, \cdot \rangle$ being the inner product and $|| \cdot ||_2$ being the Euclidean norm, $\boldsymbol{U}$ is the $N \times M$ matrix of DPD basis functions. The reference signal (the $N \times 1$ vector $\boldsymbol{y}$) is the signal to be estimated (in case of PA modeling) or to be linearized (in case of DPD linearization).

The objective is to find the $M \times 1$ vector of coefficients $\boldsymbol{p}_i$, necessary for creating a new component $\boldsymbol{U}\boldsymbol{p}_i$ maximally related to $\boldsymbol{y}$ in terms of maximal *correlation* (in the case of CCA) or maximal *covariance* (in the case of PLS). Therefore, in the case of CCA the target can be mathematically defined as

$$\max_{\boldsymbol{p}_i} \{\rho_{CCA}\} = \max_{\boldsymbol{p}_i} \left\{ \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{U}\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} \right\} \tag{7.3}$$

whereas, in the case of PLS, the target can be defined as follows,

$$\max_{\boldsymbol{p}_i} \{\rho_{PLS}\} = \max_{\boldsymbol{p}_i} \left\{ \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} \right\} \tag{7.4}$$

Note that, in case the matrix $\boldsymbol{U}$ is unitary (i.e., $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}$), then CCA is equivalent to PLS

$$\begin{aligned} \rho_{CCA} = \quad & \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{U}\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} = \quad \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{\left(\boldsymbol{p}_i^H \boldsymbol{U}^H \boldsymbol{U}\boldsymbol{p}_i\right) ||\boldsymbol{y}||_2} \\ = \quad & \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} = \quad \rho_{PLS}. \end{aligned} \tag{7.5}$$

Taking advantage of this property, at first, we use the pre-calculated PCA transformation matrix to convert the original data matrix containing the DPD basis functions into a unitary transformed matrix. Then, we apply PLS on the unitary transformed matrix. With this strategy it is able to maximize the *correlation* factor as in the CCA but using PLS which has less computational complexity. Next Section presents the DPD coefficient estimation/adaptation approach employing the combination of PCA and dynamic PLS.

7.3 The Combination of PCA and DPLS

Following a closed-loop direct learning approach (as shown in Figure 3.4), the DPD coefficients are iteratively updated as in (3.37). In a first approach, in order to reduce the number of coefficients in the DPD forward path, the greedy algorithm OMP (see Subsection 4.1.1) is applied to select the most relevant basis functions of the $N \times M$ data matrix $\boldsymbol{U}$. Then, in order to reduce the number of required DPD coefficients in the adaptation path while still keeping the same linearization performance (in comparison with QR-LS), the PCA-DPLS technique is employed. The proposed PCA-DPLS technique is described in the following.

The $M \times M$ PCA transformation matrix $\boldsymbol{P}_{pca}$ is first calculated in an off-line process. The $M \times M$ matrix $\boldsymbol{P}_{pca}$ is then used to transform the original DPD basis $\boldsymbol{U}$ into an orthogonal subspace $\widehat{\boldsymbol{U}}_{pca}$ as described in (4.14). The $N \times M$ transformed matrix $\widehat{\boldsymbol{U}}_{pca}$ contains the principal components oriented to capture the maximum variance in the data. The components in the new transformed basis $\widehat{\boldsymbol{U}}_{pca}$ are orthogonal among them and are sorted according to their relevance.

Since the PCA technique relies only in the input data, as long as the characteristics of the transmitted signals (e.g. bandwidth and power operation conditions) do not change, the obtained PCA transformation matrix $\boldsymbol{P}_{pca}$ is quite robust and thus can be calculated off-line only once and then applied to different basis $\boldsymbol{U}$. However, the matrix $\boldsymbol{P}_{pca}$ should be recalculated when the bandwidth and the power operation conditions of the signal change.

After the PCA transformation in (4.14), each column of the transformed basis $\widehat{\boldsymbol{U}}_{pca}$ is normalized as follows,

$$\bar{\boldsymbol{U}}_{pca} = \widehat{\boldsymbol{U}}_{pca}\boldsymbol{T}_{norm} = \boldsymbol{U}\boldsymbol{P}_{pca}\boldsymbol{T}_{norm} = \boldsymbol{U}\bar{\boldsymbol{P}}_{pca}. \tag{7.6}$$

$\boldsymbol{T}_{norm}$ being a $M \times M$ diagonal matrix composed by the norm of each of the columns of $\widehat{\boldsymbol{U}}_{pca}$. It is worth to mention that, like $\boldsymbol{P}_{pca}$, $\boldsymbol{T}_{norm}$ can be also calculated off-line

and only once. After the PCA transformation in (4.14) and the normalization in (7.6), the resulting transformed matrix $\bar{\boldsymbol{U}}_{pca}$ is unitary.

Then, following a dynamic PLS approach (which is an upgraded version of the DOTM algorithm presented in Subsection 6.3.1), it is possible to carry out the DPD adaptation. Algorithm 5 describes the proposed PCA-DPLS method in pseudo language. Notice that at every iteration of the adaptation, $\bar{\boldsymbol{P}}_{pca}$ is fixed, while $\boldsymbol{U}$ and $\boldsymbol{e}$ are updated with the new input data.

With the PCA-DPLS algorithm, at every iteration of DPD adaptation process we obtain a new PLS transformed matrix $\boldsymbol{P}_{dpls}$ with the minimum necessary number of new components. The criteria to decide the minimum required number of components is based on the thresholds E_{th1} and E_{th2}, defined as a percentage δ_1 and δ_2 of the energy of the error signal $\boldsymbol{e}$, respectively. The transformed basis matrix $\bar{\boldsymbol{U}}_{\boldsymbol{pca}}$ obtained from (7.6) is one more time transformed through the $M \times L$ (where L is variable and may be changed at every update iteration) transformation matrix $\boldsymbol{P}_{dpls}$ as follows

$$\widehat{\bar{\boldsymbol{U}}} = \bar{\boldsymbol{U}}_{pca}\boldsymbol{P}_{dpls} = \boldsymbol{U}\bar{\boldsymbol{P}}_{pca}\boldsymbol{P}_{dpls}. \tag{7.7}$$

The $N \times L$ transformed matrix $\widehat{\bar{\boldsymbol{U}}}$ includes L new orthonormal components which are linear combinations of the components of $\bar{\boldsymbol{U}}_{pca}$. The number of components L is variable as it depends on the dimensions of the transformation matrix $\boldsymbol{P}_{dpls}$.

If the PCA transformation matrix $\boldsymbol{P}_{pca}$ is perfectly orthogonal and the normalization matrix $\boldsymbol{T}_{norm}$ is perfectly diagonal, then the resulting transformed data matrix $\bar{\boldsymbol{U}}_{pca}$ is unitary. In this case, the combination of PCA and dynamic PLS is equivalent to CCA (as justified in Section 7.2). Therefore, after the PCA-DPLS transformation, the new orthonormal components (or columns) of the matrix $\widehat{\bar{\boldsymbol{U}}}$ appear sorted according to their contribution to maximize the correlation factor between the new components and the error signal $\boldsymbol{e}$. Having a sorted set of orthonormal components ensures that by selecting the first coefficients we get the most relevant ones in terms of correlation.

Algorithm 5 PCA-DPLS Calculation

procedure PCA-DPLS($\boldsymbol{U}, \bar{\boldsymbol{P}}_{pca}, \boldsymbol{e}, \delta_1, \delta_2$)
  initialization:
    $\boldsymbol{V}^{(0)} = \{\}$; $\boldsymbol{P}_{dpls}^{(0)} = \{\}$; $\Delta \boldsymbol{w_{pls}}^{(0)} = \{\}$;
  $j = 0$; $E_{th1} = \delta_1 ||\boldsymbol{e}||_2^2$; $E_{th2} = \delta_2 ||\boldsymbol{e}||_2^2$;
  $\boldsymbol{r}_1 = (\boldsymbol{U}\bar{\boldsymbol{P}}_{pca})^H \boldsymbol{e} = \bar{\boldsymbol{P}}_{pca}^H (\boldsymbol{U}^H \boldsymbol{e})$;
  repeat
    $j = j + 1$;
    $\boldsymbol{p}_j = \dfrac{\boldsymbol{r}_j}{||\boldsymbol{U}(\bar{\boldsymbol{P}}_{pca}\boldsymbol{r}_j)||_2}$;
    $\boldsymbol{P}_{dpls}^{(j)} \leftarrow \boldsymbol{P}_{dpls}^{(j-1)} \bigcup \boldsymbol{p}_j$;
    $\boldsymbol{v}_j = \bar{\boldsymbol{P}}_{pca}^H \boldsymbol{U}^H \boldsymbol{U} \bar{\boldsymbol{P}}_{pca} \boldsymbol{p}_j$;
    for $repeat = 1$ to 2 **do**
      for $i = 1$ to $j - 1$ **do**
        $\boldsymbol{v}_i = \boldsymbol{V}(:, i)$;
        $\boldsymbol{v}_j = \boldsymbol{v}_j - (\boldsymbol{v}_i^H \boldsymbol{v}_j)\boldsymbol{v}_i$;
      end for
    end for
    $\boldsymbol{v}_j = \dfrac{\boldsymbol{v}_j}{||\boldsymbol{v}_j||_2}$;
    $\boldsymbol{r}_j = \boldsymbol{r}_j - \boldsymbol{v}_j(\boldsymbol{v}_j^H \boldsymbol{r}_j)$;
    $\boldsymbol{V}^{(j)} \leftarrow \boldsymbol{V}^{(j-1)} \bigcup \boldsymbol{v}_j$;
    $\boldsymbol{r}_{j+1} = \boldsymbol{r}_j - \boldsymbol{V}^{(j)}((\boldsymbol{V}^{(j)})^H \boldsymbol{r}_j)$;
    $\widehat{\boldsymbol{x}}_j = \boldsymbol{U}(\bar{\boldsymbol{P}}_{pca}\boldsymbol{p}_j)$;
    $\Delta w_{pls_j} = \widehat{\boldsymbol{x}}_j^H \boldsymbol{e}$;
    $\Delta \boldsymbol{w_{pls}}^{(j)} \leftarrow \Delta \boldsymbol{w_{pls}}^{(j-1)} \bigcup \Delta w_{pls_j}$;
    $E_j = ||\Delta \boldsymbol{w_{pls}}^{(j)}||_2^2$;
  until $(|\Delta w_{pls_1}|^2 < E_{th1})$ OR $(E_j > E_{th2})$ OR $(j == dimension(\boldsymbol{U}))$
  $\Delta \widehat{\widehat{\boldsymbol{w}}} = \Delta \boldsymbol{w_{pls}}$;
  Return $\boldsymbol{P}_{dpls}, \Delta \widehat{\widehat{\boldsymbol{w}}}$
end procedure

However, if the resulting transformed data matrix $\bar{\boldsymbol{U}}_{pca}$ is not perfectly unitary, then some degradation in the sorting of the new orthonormal components will exist. In the estimation procedure, this degradation effect is detected and solved by introducing two thresholds in the coefficients extraction process. These two thresholds are set for the purpose of detecting any degradation and decide how many relevant coefficients should be calculated.

- The first threshold, E_{th1}, is determined as a percentage δ_1 of the energy (see Algorithm 5) of the error to be estimated (i.e., $E_{th1} = \delta_1 ||\boldsymbol{e}||_2^2$). The energy of the first coefficient (i.e., $|\Delta w_{pls_1}|^2$) is calculated and compared to the threshold E_{th1}. If the first coefficient is not good enough to estimate the error, the threshold is not achieved. In this case, the updating process is stopped and no more coefficients are calculated. If the first coefficient is good enough to estimate the error, then the threshold is met and more coefficients are calculated until the second threshold is reached. This is decided relying on the fact that the coefficients are sorted.

- The second threshold E_{th2} (see Algorithm 5) is determined as a percentage δ_2 of the energy of the error (i.e., $E_{th2} = \delta_2 ||\boldsymbol{e}||_2^2$). The energy of all the calculated coefficients up to iteration j (i.e., $||\boldsymbol{\Delta w_{pls}}^{(j)}||_2^2$) is evaluated and compared to the threshold E_{th2}. The PCA-DPLS algorithm will continue estimating DPD coefficients until the threshold E_{th2} is met.

The key advantage of the PCA-DPLS approach with respect to the DPLS approach adopting the DOTM algorithm presented in Subsection 6.3.1 is that the orthonormal components of the transformed matrix $\widehat{\widehat{\boldsymbol{U}}}$ are sorted. Therefore, taking advantage of this property, we can select the least number of required coefficients to achieve the given thresholds (E_{th1} and E_{th2}). Whereas with the DPLS approach, since the new components are not properly sorted, it is necessary to estimate more coefficients to reach the same threshold E_{th2}. This difference is illustrated later via

experimental results shown in Figure 7.4.

Taking into account the orthonormal property of the transformed matrix $\widehat{\widehat{\boldsymbol{U}}}$ (i.e., $\widehat{\widehat{\boldsymbol{U}}}^{H}\widehat{\widehat{\boldsymbol{U}}} = \boldsymbol{I}$), the calculation of the transformed DPD coefficient increment in (3.38) is simplified as

$$\boldsymbol{\Delta}\widehat{\widehat{\boldsymbol{w}}} = \widehat{\widehat{\boldsymbol{U}}}^{H}\boldsymbol{e} \tag{7.8}$$

Finally, the original DPD coefficient increment $\boldsymbol{\Delta w}$ can be found by applying the corresponding anti–transformations

$$\boldsymbol{\Delta w} = \bar{\boldsymbol{P}}_{pca}\boldsymbol{\Delta}\widehat{\boldsymbol{w}} = \bar{\boldsymbol{P}}_{pca}\boldsymbol{P}_{dpls}\boldsymbol{\Delta}\widehat{\widehat{\boldsymbol{w}}} \tag{7.9}$$

Figure 7.1 schematically describes the proposed PCA-DPLS approach for DPD estimation/adaptation. Experimental results proving the benefits of estimating the DPD coefficients by means of the proposed PCA-DPLS approach are shown in next Section.

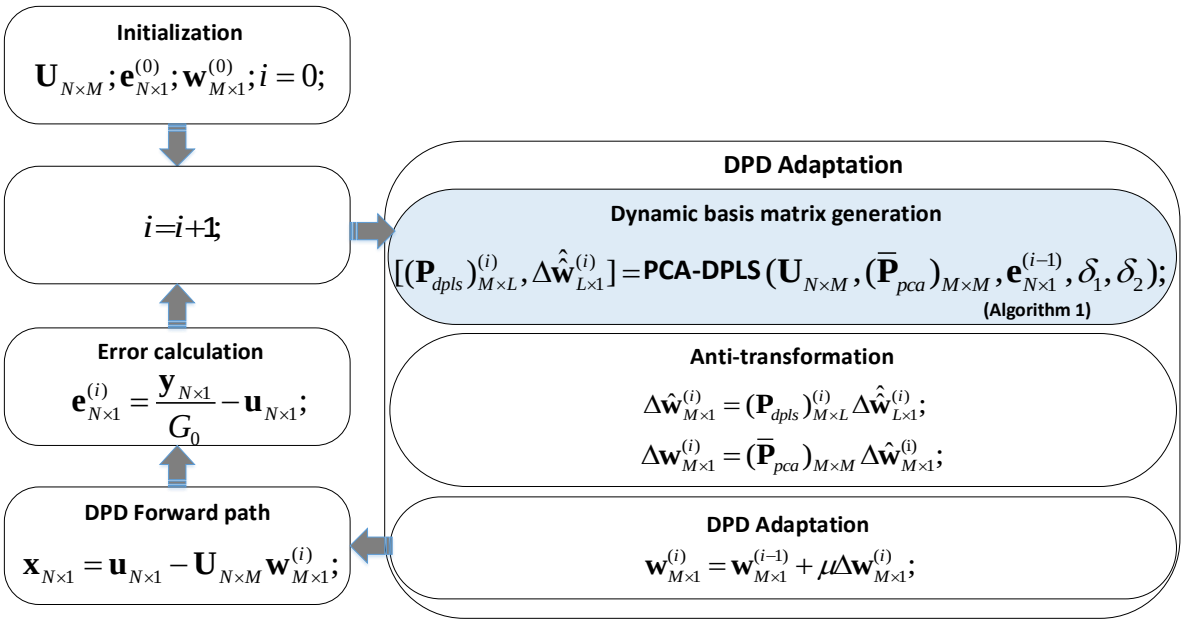


Figure 7.1: Flowchart of the DPD estimation/adaptation using the PCA-DPLS technique.

7.4 Experimental Results

In order to prove the advantages of the proposed PCA-DPLS approach for DPD coefficient estimation/adaptation, we compared it with the following techniques:

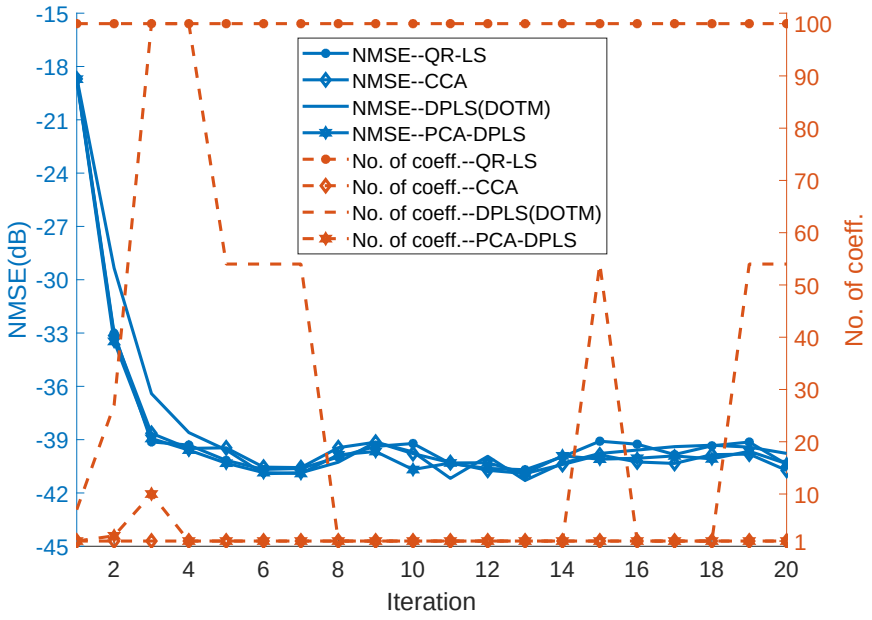
- the dynamic PLS (DOTM algorithm) introduced in Subsection 6.3.1,
- the CCA (see Subsection 4.1.2),
- the QR-LS (see Subsection 3.3.2).

The experiments are performed on the test bench described in Subsection 5.4.1. The comparison is established in terms of

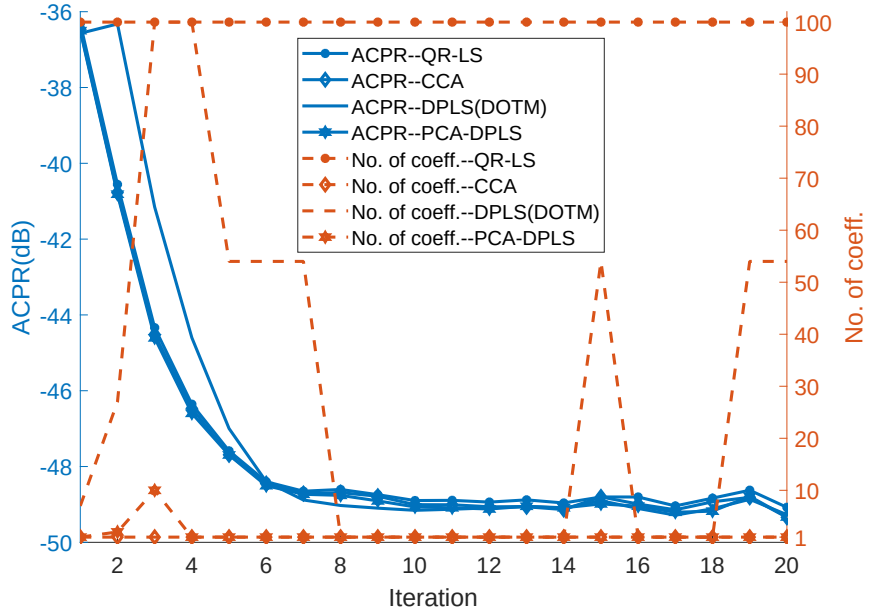
- linearization performance (via evaluating the $NMSE$ and the $ACPR$ after DPD linearization),
- the minimum number of required coefficients in the adaptation DPD subsystem to meet the linearity specifications,
- the computational running time according to Matlab's tic-toc measurements.

Firstly, the original $N \times M$ data matrix $\boldsymbol{U}$ containing the basis functions was generated using the GMP behavioral model (see Subsection 3.1.1) with 322 coefficients. The validity of the proposed PCA-DPLS method is universal, that means it does not depend on a specific PA. However, if we had used a different PA, the basis functions describing the PA behavior under some certain operating conditions (signal bandwidth, level of PA saturation) would have changed.

By applying the OMP feature selection algorithm, the number of required coefficients in the forward path are cut down to 100. Then, the tests are performed on this reduced basis matrix by applying one at a time the DPD techniques under comparison (i.e., the proposed PCA-DPLS, DPLS based on the DOTM algorithm, CCA and QR-LS).



(a)



(b)

Figure 7.2: Evolution of the $NMSE$ and $ACPR$ for PCA-DPLS, DPLS, CCA and QR-LS.

Figure 7.2a and Figure 7.2b show the $NMSE$ and $ACPR$ evolution when considering different adaptation methods and taking into account different number of

Table 7.1: DPD Performance Comparison.

Configuration 80 MHz FC-FBMC	Num. of Coeff. (max/min)	$NMSE$ [dB]	$ACPR$ [dB]	EVM [%]
No DPD	-	-18.6	-36.35	5.73
QR-LS	100/100	-40.39	-49.08	1.17
DPLS (DOTM)	100/1	-39.77	-49.27	1.19
CCA	1/1	-40.73	-49.34	1.16
PCA-DPLS	10/1	-40.35	-49.33	1.17

components (i.e. coefficients) for the DPD estimation/update. All the tested DPD techniques converge to around -40 dB of $NMSE$ after 5 iterations and -48 dB of $ACPR$ after approximately 10 iterations. However, as shown in Figure 7.2 and Table 7.1, whereas the QR-LS needs a fixed amount of coefficients (up to 100 coefficients) for each iteration of the DPD update, both DPLS (DOTM algorithm) and PCA-DPLS dynamically select the minimum necessary components to reach the targeted linearity levels. As observed in Table 7.1, the PCA-DPLS is more efficient (i.e., uses less coefficients at every iteration) than DPLS while achieving the same linearization performance. The DPLS technique dynamically selects among, for example, 100, 54, 27, 7 and 1 coefficients depending on the iteration, whereas most of the times, the PCA-DPLS needs only 1 coefficient as shown in Figure 7.2. Only when the transformed matrix $\bar{\boldsymbol{U}}_{pca}$ of the first step of the PCA-DPLS algorithm is not perfectly unitary, the algorithm needs to select more coefficients (e.g., up to 10). Finally, the CCA technique needs only 1 coefficient at every DPD adaptation iteration to meet the linearity specifications. However, as shown in Table 4.1, its computational cost is more expensive than the PLS.

Figure 7.3 shows the non-linearized spectrum and the linearized spectrum when considering 100 coefficients in the DPD forward path and 1 coefficient in the feedback path with the PCA-DPLS adaptation.

As explained in Section 7.3, after the PCA-DPLS transformation, the new or-

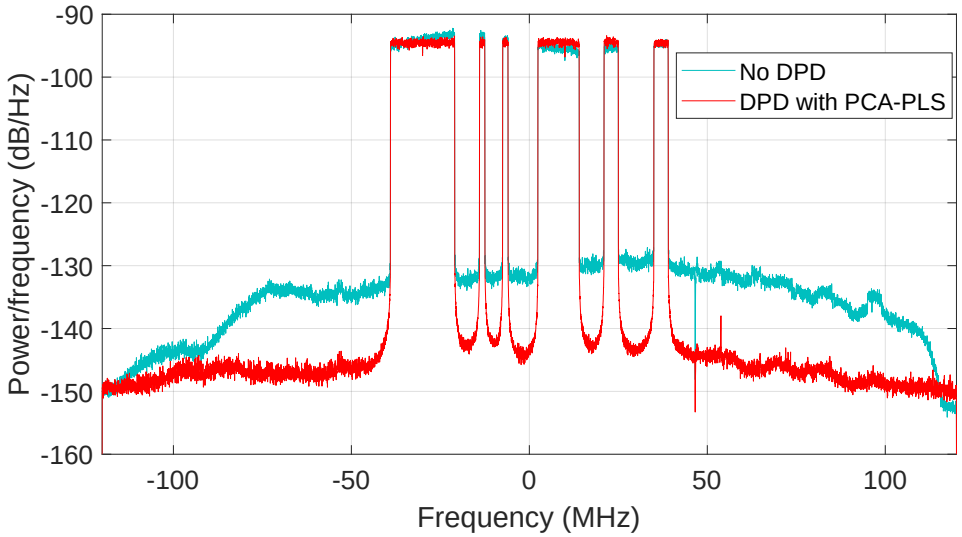


Figure 7.3: Spectra of the PA output before and after DPD linearization using the PCA-DPLS technique.

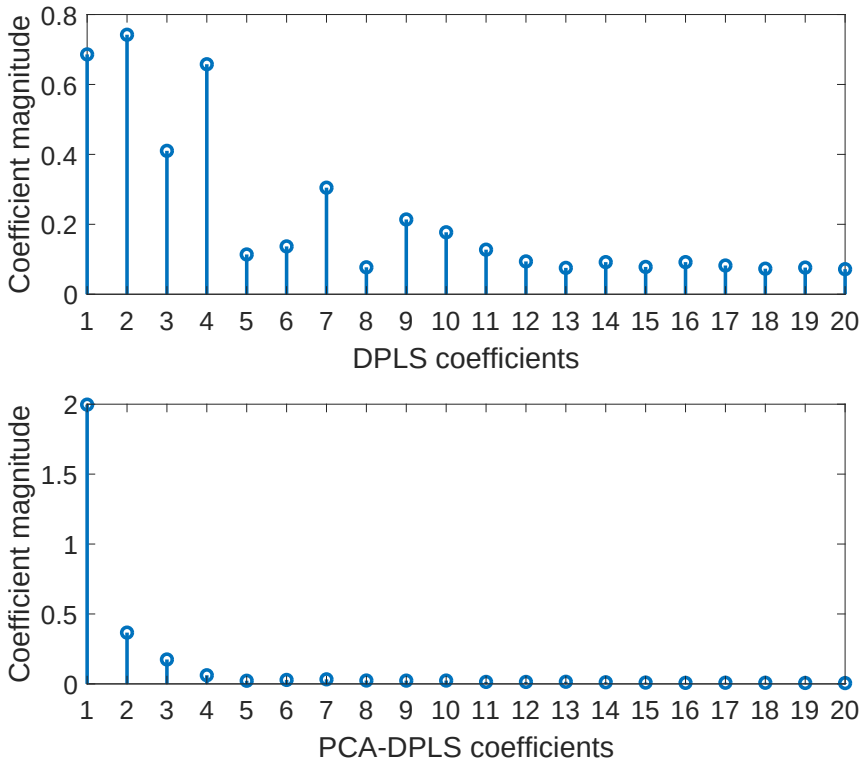


Figure 7.4: Magnitude of the DPLS (DOTM) coefficients (top) and magnitude of the PCA-DPLS coefficients (bottom) at DPD iteration update 3.

thonormal components of the matrix $\widehat{\widehat{\boldsymbol{U}}}$ (and also the coefficients) appear sorted according to their contribution to maximize the correlation factor between the new components and the error signal $\boldsymbol{e}$. This is a key advantage of the proposed PCA-DPLS with respect to DPLS. Figure 7.4 depicts the magnitude of the DPD coefficients when applying DPLS (Figure 7.4-top) and PCA-DPLS (Figure 7.4-bottom). Although in both cases the general trend shows that the magnitude of the coefficients decreases, in the case of PCA-DPLS the sorting is more accurate, i.e., close to monotonically decreasing. This allows PCA-DPLS to avoid the estimation of several unnecessary coefficients in comparison to DPLS (DOTM).

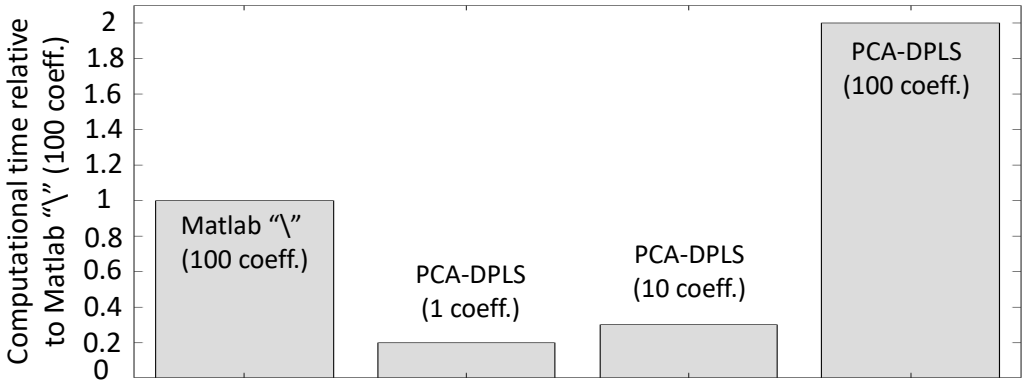


Figure 7.5: Computational time of the PCA-DPLS algorithm considering different number of coefficients and taking as a reference the computational time of Matlab's backslash operation.

In addition, the advantage of the PCA-DPLS technique is also validated in terms of the Matlab's tic-toc processing time. Taking as a reference the processing time (tic-toc) of Matlab's backslash operation with 100 coefficients, Figure 7.5 shows the relative factors of the processing time when considering PCA-DPLS using 1, 10 and 100 coefficients. It can be seen that, when considering the same number of estimated coefficients (i.e., 100 coefficients), the Matlab's backslash operation is around 2 times faster than the proposed PCA-DPLS. However, the proposed PCA-DPLS can significantly reduce the number of computed coefficients in the DPD adaptation subsystem while still achieving the same linearity levels as QR-LS. Therefore, by sig-

nificantly reducing the number of coefficients, for example, down to 10 coefficients, the PCA-DPLS processing time is only one third that of Matlab's backslash operation. Moreover, in the case of using only 1 coefficient (which happens with high probability since PCA-DPLS is equivalent to CCA when no significant degradation occurs), the PCA-DPLS running time is five times faster than Matlab's backslash operation.

7.5 Discussion

This Chapter has presented a new technique for dynamically estimating and updating the DPD coefficients based on the combination of the PCA transformation with the dynamic PLS extraction of components. The PCA-DPLS approach significantly improves the model order reduction capabilities of the DPLS technique (DOTM) presented in Section 6.3 of Chapter 6. The PCA-DPLS combination is equivalent to a CCA updating solution, which is optimal in the sense of generating components with maximum correlation. The PCA-DPLS method allows to update as many coefficients as necessary for achieving the required linearity, and to stop this update when it detects that the DPD basis is not able to estimate and minimize the remaining nonlinear error anymore. This allows to reduce the computational cost and overcome ill-conditioning problems in comparison to other methods that use a fixed number of coefficients when solving the required LS estimation in the DPD adaptation loop.

The proposed dynamic adaptation technique has been tested and compared in terms of linearization performance and computational cost with the commonly used QR decomposition approach for solving the LS problem. Experimental results show that the PCA-DPLS combination drastically reduces the amount of DPD coefficients required in the DPD adaptation subsystem while maintaining the same linearization performance, which ultimately impacts the computational cost and running time.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The emphasis of this dissertation is on the development of dimensionality reduction techniques for DPD linearization of high efficient RF PAs in wideband transmitters. The proposed techiques, BD-APCA, DPLS and PCA-DPLS, are designed by modifying and combining different feature extraction methods: PCA, PLS and CCA. The presented techniques are able to determine the minimum number of relevant DPD coefficients to be estimated and updated in the DPD feedback path. They reduce the order of the DPD behavioral models, thus achieving a reduction of the computational cost and, at the same time, ensuring a well-conditioned estimation. Moreover, the selected transformed coefficients can be estimated independently due to the orthogonality property of the new transformed basis. This simplifies remarkably the coefficient estimation in the DPD feedback path. The distinct point of the three techniques is that while BD-APCA allows an adaptive manner to estimate and update the DPD coefficients, DPLS and PCA-DPLS enable to dynamically determine the optimum number of coefficients for each update iteration. Apart from the theoretical development, all the presented methods have been simulated and experimentally tested in two instrumentation-based test beds.

The BD-APCA technique (presented in Chapter 5), based on the PCA technique,

allows to iteratively estimate and update the DPD coefficients without requiring the correlation of the DPD data matrix. The benefit of the BD-APCA to the conventional PCA is that it can be implemented in an embedded processor, such as the one available in ARM or FPGA devices. The BD-APCA algorithm is modified with respect to the CGHA algorithm in order to improve the convergence speed. Experimental results demonstrate the capability of significantly reducing the estimated DPD coefficients while still maintaining acceptable levels of linearization performance, in comparison to the full LS approach.

The DPLS method (presented in Chapter 6) dynamically estimates and updates the DPD coefficients in the feedback path. It achieves the dynamic dimensionality reduction by employing the suggested DOTM algorithm in the DPD adaptation loop. The DOTM algorithm is a modification of the SIMPLS algorithm. Experimental results show same linearity performance as QR-LS, but with a lower number of DPD coefficients.

The PCA-DPLS is a combination of PCA and DPLS, and presented in Chapter 7. It achieves equivalent dimensionality reduction to the well-known CCA method, but avoids the expensive computational cost necessary for obtaining the required canonical components. The method has been experimentally validated and compared with QR-LS, DPLS and CCA to show its advantage in terms of computational cost.

8.2 Future Directions

The research work developed in this thesis can be extended in a number of directions, as follows:

- The proper selection of the coefficients of the predistortion function in the forward path is important to achieve a desired distortion reduction with an optimum computational cost. Besides matching pursuit and OMP described and used in this dissertation, there are various efficient methods that can be ana-

lyzed and employed for selecting the most relevant DPD coefficients, such as recursive feature elimination, mutual information, and F-test [102]. Therefore, further research efforts could be focused on the DPD coefficient selection techniques in the DPD forward path.

- Since Machine Learning field of knowledge has been developed explosively over the last years, applying Machine Learning techniques to build a "smart" digital predistorter is an on-trend research direction. This dissertation presents some linearization techniques based on the concepts related to the Machine Learning field of knowledge: PCA, PLS and CCA. In the next step, more complex and effective Machine Learning structures such as artificial neural networks and deep neural networks can be employed in the DPD system to efficiently determine the DPD model and coefficients. However, their high computational complexity should be evaluated and taken into consideration.

- Nowadays, MIMO systems are popularly used in 5G networks. Improving the linearization performance and efficiency of MIMO systems is an attractive topic [52,103,104]. Unlike in SISO systems, in MIMO systems, cross-interference and other unwanted effects have to be considered, thus demanding additional complexity of the predistorter. Therefore, the high computational complexity/cost of the predistorter for MIMO systems deserves our attention, as a field where to apply the presented methods for dimensionality reduction.

List of Algorithms

List of Figures

List of Tables

Bibliography

[1] J. Wu, Y. Zhang, M. Zukerman, and E. K.-N. Yung, "Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 803–826, 2015.

[2] H. Chireix, "High Power Outphasing Modulation," *Proceedings of the IRE*, vol. 23, no. 11, pp. 1370–1392, Nov. 1935.

[3] W. H. Doherty, "A new high efficiency power amplifier for modulated waves," *Radio Engineers, Proceedings of the Institute of*, vol. 24, no. 9, pp. 1163–1182, 1936.

[4] F. Wang, A. Yang, D. Kimball, L. Larson, and P. Asbeck, "Design of wide-bandwidth envelope-tracking power amplifiers for OFDM applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 4, pp. 1244–1255, apr 2005.

[5] L. R. Kahn, "Single-Sideband Transmission by Envelope Elimination and Restoration," *Proceedings of the IRE*, vol. 40, no. 7, pp. 803–806, July 1952.

[6] R. Quaglia and S. Cripps, "A load modulated balanced amplifier for telecom applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 3, pp. 1328–1338, March 2018.

[7] P. Lavrador, T. Cunha, P. Cabral, and J. Pedro, "The linearity-efficiency compromise," *IEEE Microwave Magazine*, vol. 11, no. 5, pp. 44–58, aug 2010.

[8] A. Katz, J. Wood, and D. Chokola, "The evolution of PA linearization: From classic feedforward and feedback through analog and digital predistortion," *IEEE Microwave Magazine*, vol. 17, no. 2, pp. 32–40, feb 2016.

[9] R. N. Braithwaite, *Digital Processing for Front End in Wireless Communication and Broadcasting*. Cambridge University Press, 2011, ch. General principles and design overview of digital predistortion, pp. 143–191.

[10] R. Braithwaite, "Low cost, low delay UMTS power amplifier using digital-controlled adaptive analog predistortion," in *2006 European Conference on Wireless Technologies*. IEEE, sep 2006.

[11] P. L. Gilabert and G. Montoro, "3-D Distributed Memory Polynomial Behavioral Model for Concurrent Dual-Band ET PA Linearization," *IEEE Trans. on Microw. Theory and Tech.*, vol. 63, pp. 638–648, Feb 2015.

[12] D. H. Wisell, "Exploring the sampling rate requirements for behavioural amplifier modelling," in *XVIII IMEKO World Congress*, Sept. 2006, pp. 1–4.

[13] P. L. Gilabert, "Multi Look-Up Table Digital Predistortion for RF Power Amplifier Linearization," Doctoral Thesis in Signal Theory and Communications, Universitat Politècnica de Catalunya, 2007.

[14] D. Schreurs, M. ODroma, A. A. Goacher, and M. Gadringer, Eds., *RF Power Amplifier Behavioral Modeling*. Cambridge University Press, 2008.

[15] F. H. Raab, P. Asbeck, S. Cripps, P. B. Kenington, Z. B. Popovic, N. Pothecary, J. F. Sevic, and N. O. Sokal, "Power Amplifiers and Transmitters for RF and Microwave," *IEEE Trans. on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 814–826, March 2002.

[16] D. Cox, "Linear amplification with nonlinear components," *IEEE Transactions on Communications*, vol. 22, no. 12, pp. 1942–1945, dec 1974.

[17] M. El-Asmar, A. Birafane, M. Helaoui, A. B. Kouki, and F. M. Ghannouchi, "Analytical design methodology of outphasing amplification systems using a new simplified chireix combiner model," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 6, pp. 1886–1895, jun 2012.

[18] S. Hu, S. Kousai, J. S. Park, O. L. Chlieh, and H. Wang, "Design of a transformer-based reconfigurable digital polar doherty power amplifier fully integrated in bulk CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 5, pp. 1094–1106, may 2015.

[19] R. Pengelly, C. Fager, and M. Ozen, "Doherty's legacy: A history of the doherty power amplifier from 1936 to the present day," *IEEE Microwave Magazine*, vol. 17, no. 2, pp. 41–58, feb 2016.

[20] C. Ma, W. Pan, S. Shao, C. Qing, and Y. Tang, "A wideband doherty power amplifier with 100 MHz instantaneous bandwidth for LTE-advanced applications," *IEEE Microwave and Wireless Components Letters*, vol. 23, no. 11, pp. 614–616, nov 2013.

[21] J. H. Kim, G. D. Jo, J. H. Oh, Y. H. Kim, K. C. Lee, J. H. Jung, and C. S. Park, "High-efficiency envelope-tracking transmitter with optimized class-F amplifier and 2-bit envelope amplifier for 3g LTE base station," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 6, pp. 1610–1621, jun 2011.

[22] M. Hassan, L. Larson, V. Leung, D. Kimball, and P. Asbeck, "A wideband cmos/gaas hbt envelope tracking power amplifier for 4g lte mobile terminal applications," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 60, no. 5, pp. 1321–1330, 2012.

[23] G. Montoro, P. Gilabert, E. Bertran, and J. Berenguer, "A method for real-time generation of slew-rate limited envelopes in envelope tracking transmitters," in *RF Front-ends for Software Defined and Cognitive Radio Solutions (IMWS), 2010 IEEE International Microwave Workshop Series on*, 2010, pp. 1–4.

[24] F. Mazlumi and M. Khoshgard, "Modulation error ratio degradation due to error sources existing in a DRM transmitter using envelope elimination and restoration," in *2006 European Conference on Wireless Technologies*. IEEE, sep 2006.

[25] P. Fedorenko and J. S. Kenney, "Analysis and suppression of memory effects in envelope elimination and restoration (EER) power amplifiers," in *2007 IEEE/MTT-S International Microwave Symposium*. IEEE, jun 2007.

[26] A. Mamdouh, M. Aboudina, F. Hussien, and A. N. Mohieldin, "Efficient supply modulator for wide-band envelope elimination and restoration power amplifiers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2019.

[27] X. Ruan, "A review of envelope tracking power supply for mobile communication systems," *CPSS Transactions on Power Electronics and Applications*, vol. 2, no. 4, pp. 277–291, dec 2017.

[28] R. Kline, "Harold Black and the negative-feedback amplifier," *IEEE Control Systems*, vol. 13, no. 4, pp. 82–85, aug 1993.

[29] T. Qi and S. He, "Power up potential power amplifier technologies for 5g applications," *IEEE Microwave Magazine*, vol. 20, no. 6, pp. 89–101, jun 2019.

[30] *https://www.invent.org/inductees/harold-stephen-black*.

[31] A. A. M. Saleh and J. Salz, "Adaptive linearization of power amplifiers in digital radio systems," *Bell System Technical Journal*, vol. 62, no. 4, pp. 1019–1033, apr 1983.

[32] C. Eun and E. Powers, "A new volterra predistorter based on the indirect learning architecture," *IEEE Transactions on Signal Processing*, vol. 45, no. 1, pp. 223–227, 1997.

[33] S. Andreoli, H. G. McClure, P. Banelli, and S. Cacopardi, "Digital Linearizer for RF Amplifiers," *IEEE Trans. on Broadcasting*, vol. 43, no. 1, pp. 12–19, March 1997.

[34] L. Ding, G. Zhou, D. Morgan, Z. Ma, J. Kenney, J. Kim, and C. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Transactions on Communications*, vol. 52, no. 1, pp. 159–165, jan 2004.

[35] G. Montoro, P. L. Gilabert, E. Bertran, A. Cesari, and D. D. Silveira, "A New Digital Predictive Predistorter for Behavioral Power Amplifier Linearization," *IEEE Microwave and Wireless Components Letters*, vol. 17, no. 6, pp. 448–450, June 2007.

[36] A. Katz, R. Gray, and R. Dorval, "Truly wideband linearization," *IEEE Microwave Magazine*, vol. 10, no. 7, pp. 20–27, dec 2009.

[37] M. Cho, "Analog Predistortion for Improvement of RF Power Amplifier Efficiency and Linearity," Doctoral Thesis in Electrical and Computer Engineering, Georgia Institute of Technology, 2016.

[38] D. Scheurs, M. O'Droma, A. A. Goacher, and M. Gadringer, Eds., *RF Power Amplifier Behavioural Modeling*. Cambridge University Press, 2009.

[39] V. Volterra, *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Phoenix Editions, 1959.

[40] D. Saffar, N. Boulejfen, F. M. Ghannouchi, A. Gharsallah, and M. Helaoui, "Behavioral modeling of MIMO nonlinear systems with multivariable polynomials," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 11, pp. 2994–3003, nov 2011.

[41] S. Amin, P. N. Landin, P. Handel, and D. Ronnow, "Behavioral modeling and linearization of crosstalk and memory effects in RF MIMO transmitters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 4, pp. 810–823, apr 2014.

[42] A. Abdelhafiz, L. Behjat, F. M. Ghannouchi, M. Helaoui, and O. Hammi, "A high-performance complexity reduced behavioral model and digital predistorter for MIMO systems with crosstalk," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 1996–2004, may 2016.

[43] A. Zhu and T. J. Brazil, "Behavioral modeling of rf power amplifiers based on pruned volterra series," vol. 14, no. 12, pp. 563–565, 2004.

[44] A. Zhu, J. C. Pedro, and T. J. Brazil, "Dynamic deviation reduction-based volterra behavioral modeling of RF power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 12, pp. 4323–4332, dec 2006.

[45] A. Zhu, J. C. Pedro, and T. R. Cunha, "Pruning the volterra series for behavioral modeling of power amplifiers using physical knowledge," *IEEE Transactions on Microwave Theory and Techniques*, vol. 55, no. 5, pp. 813–821, may 2007.

[46] A. Melendez-Cano, S. A. Juarez-Cazares, J. A. Galaviz-Aguilar, J. R. Cardenas-Valdez, M. J. Garcia-Ortega, A. Calvillo-Tellez, P. Roblin, and J. C. Nunez-Perez, "Behavioral modeling for power amplifiers comparing MPM, wiener and hammerstein with FPGA-based implementation," in *2016 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*. IEEE, nov 2016.

[47] J. Kim and K. Konstantinou, "Digital Predistortion of Wideband Signals Based on Power Amplifier Model with Memory," in *Electronics Letters*, vol. 37, no. 23, Nov. 2001, pp. 1417–1418.

[48] D. R. Morgan, Z. Ma et al., "A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers," *IEEE Trans. on Signal Processing*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[49] G. Montoro, P. L. Gilabert, E. Bertran, A. Cesari, and D. D. Silveira, "A New Digital Predictive Predistorter for Behavioral Power Amplifier Linearization," *IEEE Microwave and Wireless Components Letters*, vol. 17, no. 6, pp. 448–450, June 2007.

[50] P. Roblin, C. Quindroit, N. Naraharisetti, S. Gheitanchi, and M. Fitton, "Concurrent linearization: The state of the art for modeling and linearization of multiband power amplifiers," *IEEE Microwave Magazine*, vol. 14, no. 7, pp. 75–91, nov 2013.

[51] P. L. Gilabert and G. Montoro, "3-D distributed memory polynomial behavioral model for concurrent dual-band envelope tracking power amplifier linearization," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 2, pp. 638–648, Feb. 2015.

[52] K. Hausmair, P. N. Landin, U. Gustavsson, C. Fager, and T. Eriksson, "Digital predistortion for multi-antenna transmitters affected by antenna crosstalk," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 3, pp. 1524–1535, mar 2018.

[53] L. Guan and A. Zhu, "Low-cost fpga implementation of volterra series-based digital predistorter for rf power amplifiers," vol. 58, no. 4, pp. 866–872, 2010.

[54] S.-L. Cheng, W.-R. Wu, C.-H. Peng, C.-J. Hsu, and P. Liang, "Digital predistortion for concurrent dual-band transmitter using a 2d LUT based method,"

in *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*. IEEE, apr 2015.

[55] A. Kwan, F. Ghannouchi, O. Hammi, M. Helaoui, and M. Smith, "Look-up table-based digital predistorter implementation for field programmable gate arrays using long-term evolution signals with 60 MHz bandwidth," *IET Science, Measurement & Technology*, vol. 6, no. 3, p. 181, 2012.

[56] Y. Ma, Y. Yamao, Y. Akaiwa, and C. Yu, "FPGA implementation of adaptive digital predistorter with fast convergence rate and low complexity for multi-channel transmitters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 11, pp. 3961–3973, nov 2013.

[57] A. Molina, K. Rajamani, and K. Azadet, "Concurrent dual-band digital predistortion using 2-D lookup tables with bilinear interpolation and extrapolation: Direct least squares coefficient adaptation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 4, pp. 1381–1393, Apr. 2017.

[58] ——, "Digital predistortion using lookup tables with linear interpolation and extrapolation: Direct least squares coefficient adaptation," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 3, pp. 980–987, Mar. 2017.

[59] L. N. Trefethen and D. B. III, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, USA, 1997.

[60] S. Stapleton, G. Kandola, and J. Cavers, "Simulation and analysis of an adaptive predistorter utilizing a complex spectral convolution," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 387–394, 1992.

[61] P. L. Gilabert, A. Cesari, G. Montoro, E. Bertran, and J. M. Dilhac, "Multi Look-Up Table FPGA Implementation of a Digital Adaptive Predistorter for Linearizing RF Power Amplifiers with Memory Effects," *IEEE Trans. on Microwave Theory and Techniques*, p. Submitted for revision, 2007.

[62] N. Mrabet, I. Mohammad, F. Mkadem, C. Rebai, and S. Boumaiza, "Optimized hardware for polynomial digital predistortion system implementation," in *2012 IEEE Topical Conf. on Power Amplifiers for Wireless and Radio Appl. (PAWR)*. IEEE, Jan. 2012, pp. 83–84.

[63] W. Cao, Y. Li, and A. Zhu, "Magnitude-selective affine function based digital predistorter for RF power amplifiers in 5g small-cell transmitters," in *2017 IEEE MTT-S International Microwave Symposium (IMS)*. IEEE, jun 2017.

[64] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A Robust Digital Baseband Predistorter Constructed Using Memory

Polynomials," *IEEE Trans. on Communications*, vol. 52, no. 1, pp. 159–165, Jan 2004.

[65] R. N. Braithwaite, "Closed-loop digital predistortion (DPD) using an observation path with limited bandwidth," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 2, pp. 726–736, feb 2015.

[66] ——, "Digital predistortion of an RF power amplifier using a reduced volterra series model with a memory polynomial estimator," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 10, pp. 3613–3623, Oct. 2017.

[67] D. Lopez-Bueno, Q. A. Pham, G. Montoro, and P. L. Gilabert, "Independent digital predistortion parameters estimation using adaptive principal component analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5771–5779, dec 2018.

[68] H. Paaso and A. Mammela, "Comparison of direct learning and indirect learning predistortion architectures," in *2008 IEEE International Symposium on Wireless Communication Systems*. IEEE, oct 2008.

[69] R. N. Braithwaite, "A comparison of indirect learning and closed loop estimators used in digital predistortion of power amplifiers," in *2015 IEEE MTT-S Int. Microw. Symp.*, May 2015.

[70] D. Zhou and V. E. DeBrunner, "Novel adaptive nonlinear predistorters based on the direct learning algorithm," *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 120–133, jan 2007.

[71] A. N. Tikhonov and V. Y. Arsensin, *Solution of ill-posed problems*. V H Winston, Washington DC, 1977.

[72] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[73] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, apr 2005.

[74] D. Wisell, J. Jalden, and P. Handel, "Behavioral power amplifier modeling using the lasso," in *2008 IEEE Inst. and Meas. Tech. Conf.*, May 2008, pp. 1864–1867.

[75] L. Guan and A. Zhu, "Optimized Low-Complexity Implementation of Least Squares Based Model Extraction for Digital Predistortion of RF Power Amplifiers," *IEEE Trans. on Microw. Theory and Tech.*, vol. 60, no. 3, pp. 594–603, Jan. 2012.

[76] S. Mallat and Z. Zhang, "Adaptive time-frequency decomposition with matching pursuits," in *[1992] Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*. IEEE.

[77] J. Reina-Tosina, M. Allegue et al., "Behavioral modeling and predistortion of power amplifiers under sparsity hypothesis," *IEEE Trans. on Microw. Theory and Tech.*, vol. 63, no. 2, pp. 745–753, Feb. 2015.

[78] P. L. Gilabert, G. Montoro, D. Lopez, and J. A. Garcia, "3D Digital predistortion for dual-band envelope tracking power amplifiers," in *Microwave Conference Proceedings (APMC), 2013 Asia-Pacific*, 2013, pp. 734–736.

[79] P. L. Gilabert, G. Montoro, et al., "Comparison of model order reduction techniques for digital predistortion of power amplifiers," in *46th European Microw. Conf. (EuMC)*, Oct. 2016, pp. 182–185.

[80] H. Hotelling, "Relations Betwen Two Sets of Variates," *Biometrika*, vol. 28, no. 3-4, pp. 321–377, dec 1936.

[81] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, feb 1970.

[82] R. Tibshirani, I. Johnstone, T. Hastie, and B. Efron, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, apr 2004.

[83] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[84] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*. IEEE Comput. Soc. Press.

[85] A.-K. Seghouane, "Asymptotic bootstrap corrections of aic for linear regression models," *Signal Process.*, vol. 90, no. 1, pp. 217–224, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2009.06.010

[86] *Principal Component Analysis*. Springer-Verlag, 2002.

[87] H. Wold, "Partial least squares," *Encyclopedia of Statistical Sciences*, vol. 6, pp. 581–591, 1985.

[88] A. Sharma and K. K. Paliwal, "Fast principal component analysis using fixed-point algorithm," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1151–1155, jul 2007.

[89] F. Westad, K. Diepold, and H. Martens, "QR-PLSR: reduced-rank regression for high-speed hardware implementation," *Journal Of Chemometrics*, vol. 10, pp. 439–451, 1996.

[90] B. V. Srinivasan, W. R. Schwartz, R. Duraiswami, and L. Davis, "Partial least squares on graphical processor for efficient pattern recognition," Tech. Rep., 2010.

[91] J. Chen, G. Wang, Y. Shen, and G. B. Giannakis, "Canonical correlation analysis of datasets with a common source graph," *IEEE Transactions on Signal Processing*, vol. 66, no. 16, pp. 4398–4408, aug 2018.

[92] D. Wisell, J. Jalden, and P. Handel, "Behavioral power amplifier modeling using the lasso," in *2008 IEEE Inst. and Meas. Tech. Conf.*, May 2008, pp. 1864–1867.

[93] L. Guan and A. Zhu, "Optimized low-complexity implementation of least squares based model extraction for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 3, pp. 594–603, Mar. 2012.

[94] J. Peng, S. He, B. Wang, Z. Dai, and J. Pang, "Digital predistortion for power amplifier based on sparse bayesian learning," *IEEE Trans. on Circ. and Sys. II: Express Briefs*, vol. 63, no. 9, pp. 828–832, Sep. 2016.

[95] S. de Jong, "SIMPLS: An alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, Mar. 1993.

[96] X. Hu and K. Rao, "Error concealment for video transmission using adaptive principal component analysis with missing data," in *2005 International Symposium on Intelligent Signal Processing and Communication Systems*. IEEE, 2005.

[97] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, no. 6, pp. 459–473, jan 1989.

[98] Y. Zhang and Y. Ma, "CGHA for principal component extraction in the complex domain," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 1031–1036, 1997.

[99] Q. A. Pham, D. Lopez-Bueno, T. Wang, G. Montoro, and P. L. Gilabert, "Multidimensional LUT-based digital predistorter for concurrent dual-band envelope tracking power amplifier linearization," in *2018 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR)*. IEEE, jan 2018.

[100] S. Haykin, *Neural Networks and Learning Machines*. 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2009.

[101] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection*. Springer Berlin Heidelberg, 2006, pp. 34–51.

[102] A. M. Pirbazari, A. Chakravorty, and C. Rong, "Evaluating feature selection methods for short-term load forecasting," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, feb 2019.

[103] P. Jaraut, M. Rawat, and F. M. Ghannouchi, "Composite neural network digital predistortion model for joint mitigation of crosstalk, i/q imbalance, nonlinearity in MIMO transmitters," *IEEE Transactions on Microwave Theory and Techniques*, pp. 1–10, 2018.

[104] X. Liu, Q. Zhang, W. Chen, H. Feng, L. Chen, F. M. Ghannouchi, and Z. Feng, "Beam-oriented digital predistortion for 5g massive MIMO hybrid beamforming transmitters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 7, pp. 3419–3432, jul 2018.